# PLAYING SNAKE ON A GRAPH

DENISE GRAAFSMA

A Master's Thesis
Supervised by Bodo Manthey and Alexander Skopalik

University of Twente
Enschede, The Netherlands
August 2024

**Abstract**

The mathematical study of puzzles and games has gained quite some popularity. We contribute to this growing area of research by introducing the game of Snake on a graph. Based on the classic computer game Snake, a snake forms a simple path that has to move to an apple while avoiding colliding with itself. When the snake reaches the apple, it grows longer, and a new apple appears. A graph on which the snake has a strategy to keep eating apples until it covers all the vertices of the graph is called snake-winnable. We refer to the problem of determining whether a graph is snake-winnable as the snake problem.

We prove the snake problem is NP-hard, even when restricted to grid graphs. For odd-sized bipartite graphs and graphs with vertex-connectivity 1, we fully characterize the snake-winnable graphs. Furthermore, we show that non-Hamiltonian graphs with a girth greater than 6 are never snake-winnable and provide a necessary graph structure for all snake-winnable graphs.

# Contents

# 1. Introduction

## 1.1 The game of Snake

Many will remember playing Snake on their mobile phone. Created by Taneli Armanto [7], its introduction on Nokia phones in 1998 popularized mobile phone games, and solidified it as a timeless classic. Players control a snake and have to guide it to apples. With each apple consumed, the snake grows longer. The challenge is to grow the snake as long as possible while avoiding collisions with its own body or the screen's borders.

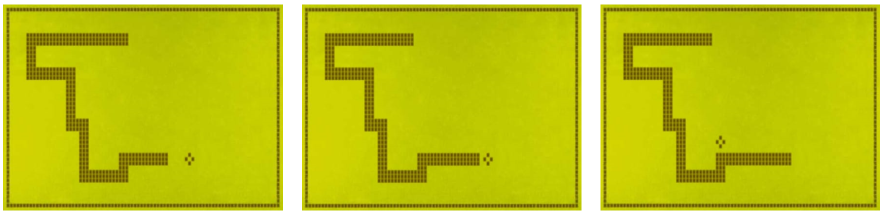


Figure 1.1: A recreation of the 1998 Nokia snake game.

The origin of Snake, however, dates back much further than the Nokia version. In 1976 the arcade game Blockade was released by Gremlin Industries [6]. Unlike the Nokia version, this was a two-player game where both players controlled a snake and had to block each other.

A single-player version called Snake Byte was published in 1982 by Sirius Software [9]. Written by Chuck Sommerville, the game was already quite similar to the Nokia version but also featured solid obstacles and moving plums that the snake had to avoid. The game Nibbler, also released in 1982 [8], focused more heavily on the obstacles and had the snake navigate a maze-like structure to reach the apples. Feeling more like a hybrid between snake and Pac-Man, the game would start with all the apples already placed within the maze, and the snake had to consume them all to complete the level.

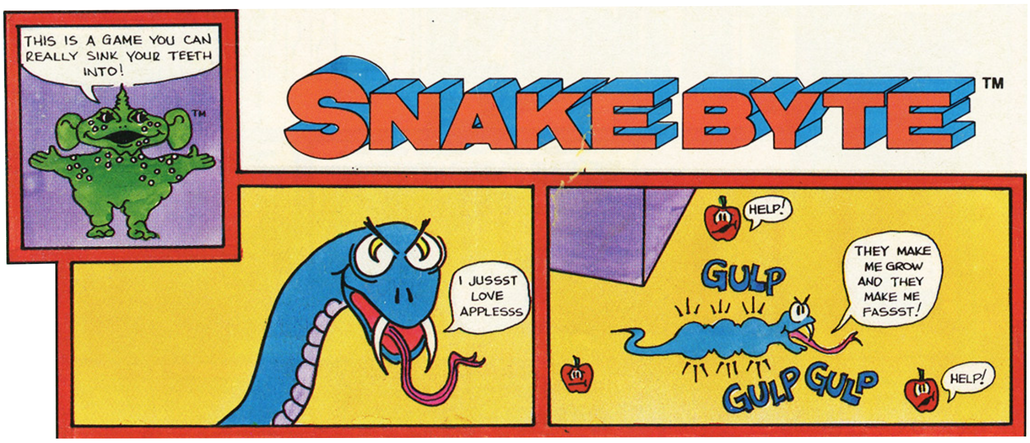



Figure 1.2: Part of an advertisement for Snake Byte from the Summer 1983 Sirius Product Catalog.

Over the years, many new versions of the game have been released. A search for "snake" on Google even provides you with a playable version right on the search results page. Inspired by this popular game, we will continue the tradition of turning fun games into even more exciting mathematics.

Since the game of snake is usually played on a grid, it has a natural translation to a game on a grid graph. The snake forms a path on the graph and moves from vertex to vertex to navigate to a vertex with an apple. But we do not have to restrict ourselves to grid graphs since this game can be generalized to any connected graph.

## 1.2 Related work

Over the years, many puzzles and games have been studied through a mathematical lens. Demaine provides an extensive overview of the area of combinatorial games [4], including games such as Checkers and Jenga, and puzzles like Sudoku and Minesweeper.

Closer related to our game of Snake, is the work by De Biasi and Ophelders [3] on the Nibbler food collection problem. This problem asks the following: given a graph, the food locations, growth rate, and starting position of the snake, can the snake collect all the food? The growth rate indicates how much the snake grows each time it eats a piece of food. The Nibbler food collection problem is NP-hard, even when restricted to solid grid graphs. Furthermore, if the growth rate is at least 2, then it is also NP-hard on rectangular grid graphs.

## 1.3 Our contributions

While the work by De Biasi and Ophelders is primarily inspired by the game Nibbler, our version of Snake on a graph is closer to the Nokia version. Instead of having all the apples on the graph from the beginning, only one apple is present at a time. A new apple is placed only when the previous one is consumed, meaning the snake does not know the locations of future apples. Hence, it will have to adjust its strategy according to where the next apple appears. In contrast to the game described by De Biasi and Ophelders, which is framed as a motion planning problem, our game can be viewed as a two-player game where the apple placer acts as an adversary to the snake. We also generalize the game to be played on any connected graph, rather than strictly adhering to the original game of Snake and only considering grid graphs.

For Snake on a graph, we show that determining whether the snake has a winning strategy is NP-hard, even when restricted to grid graphs (Section 4). We also take steps towards characterizing the graphs with a winning strategy for the snake. We give a full characterization of the winnable odd-sized bipartite graphs (Section 3.2.1) and winnable graphs with vertex-connectivity 1 (Section 7.1). Furthermore, we show that the snake can never win on non-Hamiltonian graphs with a girth greater than 6 (Section 5). Finally, we provide a necessary graph structure for the snake to have a winning strategy (Section 6).

## 1.4 Preliminaries and notation

We provide a brief summary of the notation used, as well as some of the requisite background. For a more in-depth source on graph theory, we refer to the book *Graph Theory With Applications* by Bondy and Murty [1].

For a graph $G$, we write $G = (V, E)$, where $V$ is the set over *vertices* and $E$ is the set of *edges* of $G$. By $|V|$ and $|E|$ we denote the cardinality of these sets. An edge between the vertices $u$ and $v$ is denoted $uv$. For a vertex $v \in V$, the *neighbor set* is denoted $N(v)$ and consists of all vertices that are adjacent to $v$. The *degree* of $v$, denoted $d(v)$ is the cardinality of $N(v)$.

Let $G'$ be a subgraph of $G$. By $V(G')$ we denote the set of vertices and by $E(G')$ the set of edges of $G'$. A subgraph $G'$ is *complete* if $uv \in E(G')$ for all $u, v \in V(G')$. Otherwise, we call $G'$ *incomplete*. For some $v \in V \backslash V(G')$, $G' + v$ denotes the graph obtained by adding $v$, and all the edges $uv$ with $u \in V(G')$ to $G'$.

Let $S$ be a subset of $V$. The subgraph *induced* by $S$ consists of the vertices of $S$, and all the edges of $E$ with both endpoints in $S$. By $G - S$ we denote the subgraph of $G$ induced by $V \backslash S$. If $S = \{v\}$, then by slight abuse notation we denote $G - S = G - v$.

A *path* $P = (p_1, \ldots, p_k)$ is an ordered set of vertices with $p_i p_{i+1} \in E$ for all $i \in \{1, \ldots, k-1\}$. The vertices $p_1$ and $p_k$ are the *endpoints*, and all other vertices are *internal vertices* of $P$. The path is *simple* if no two vertices on the path are the same. In some cases, we will view $P$ as the ordered set of edges $(p_1p_2, p_2p_3, \ldots p_{k-1}p_k)$. By $|P|$, we denote the number of edges on $P$, which we call the *length* of the path. Note the number of vertices on a path is always 1 more than the length of the path. We denote $\overline{P} = V \backslash V(P)$.

If for every $u, v \in V$, there exists a path from between $u$ and $v$, then we say $G$ is *connected*. Otherwise, we call $G$ *disconnected*.

A *cycle* $C = (c_1, \ldots, c_k)$ is a path with $c_k c_1 \in E$. When considering the cycle as a set of edges, we also view $c_k c_1$ as part of the cycle. Hence, the length of a cycle $|C|$ denotes the number of edges, as well as the number of vertices on $C$. Just like paths, a cycle is *simple* if no two vertices on the cycle are the same, and $\overline{C} = V \backslash V(C)$.

A *Hamiltonian cycle* in $G = (V, E)$ is a simple cycle that contains all the vertices of $V$. If a graph has a Hamiltonian cycle, then we call it *Hamiltonian*. Otherwise, refer to it as *non-Hamiltonian*.

A graph $G = (V, E)$ is *bipartite* if $V$ can be partitioned into two disjoint sets $X$ and $Y$ such that every edge has one endpoint in $X$ and the other in $Y$.

If a graph can be embedded in the plane, that is, it can be drawn in the plane without any crossing edges, then it is *planar*. The *outer face* of an embedding is the unbounded region that surrounds the graph. The *outer boundary* consists of all the edges and vertices that directly border the outer face.

# 2. The game of Snake on a graph

In the game of Snake, a snake moves around a grid while avoiding colliding with the surrounding walls or its own body. The snake consists of a sequence of grid squares. The first square of this sequence, the head of the snake, can move to any of the free adjacent squares. The rest of the body follows the head.

At any point in time, one of the grid squares contains an apple. The snake's aim is to eat this apple by moving its head onto this square. When the snake eats the apple, it grows one square longer. As this happens, a new apple appears on one of the free squares. An example of the gameplay can be found in Figure 2.1.

Part of the difficulty of the game lies in the fact that the snake is always moving, requiring the player to react quickly. In many versions, the snake even speeds up as the game moves on. We will, however, completely eliminate this element. We are not interested in whether the player is quick enough to make a certain move; instead, we want to know whether the move is possible at all.

Since the game of Snake is played on a grid, we can describe it as a game on a grid graph, as depicted in Figure 2.2. The snake forms a simple path that moves around in the graph. While the original game would always be played on a grid graph, we can generalize it to any type of simple graph.

Let us now formally define the rules of Snake on a graph. The game is played on a connected simple graph $G = (V, E)$ with $|V| \geq 3$. From now on, we will assume all our graphs have this property.

## 2.1 Snake position

During the game, the snake will always occupy an ordered set of vertices, which must form a simple path. We define the *length* of the snake as the number of vertices on this path. Since the length of a path is defined as the number of edges it contains, the length of the snake will always be one more than the length of the path it forms.

Let $\ell$ be the current length of the snake, then we denote the position of the snake by $S = (s_1, \dots, s_\ell)$. We will refer to $s_1$ as the head, and $s_\ell$ as the tail of the snake. By $\overline{S}$, we denote all vertices in $V$ that are not on $S$. We will refer to $\overline{S}$ as the *unoccupied set*.

In some cases, we will index the snake's position by time to better describe the snake's movement. By $S^t = (s_1^t, \dots, s_\ell^t)$, we denote the position of the snake at time $t$.

## 2.2 Snake movement

Let $S^t = (s_1^t, \dots, s_\ell^t)$ be the current position of the snake. For a vertex $v \in V$, the neighbor set $N(v)$ denotes the set of vertices in $V$ that are adjacent to $v$. The head of the snake must move to a vertex in $N(s_1^t)$. Suppose the head moves from $s_1^t$ to some vertex $v \in N(s_1^t)$.

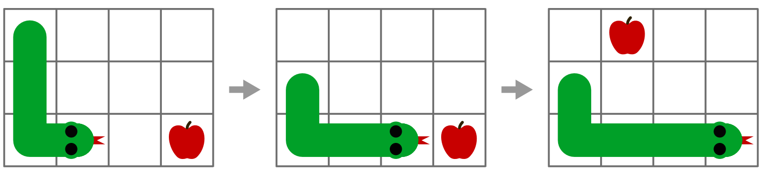


Figure 2.1: The snake moves towards the apple and grows by one square.

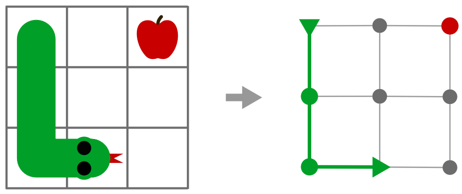


Figure 2.2: The game of Snake can be described as a game on a rectangular grid graph.

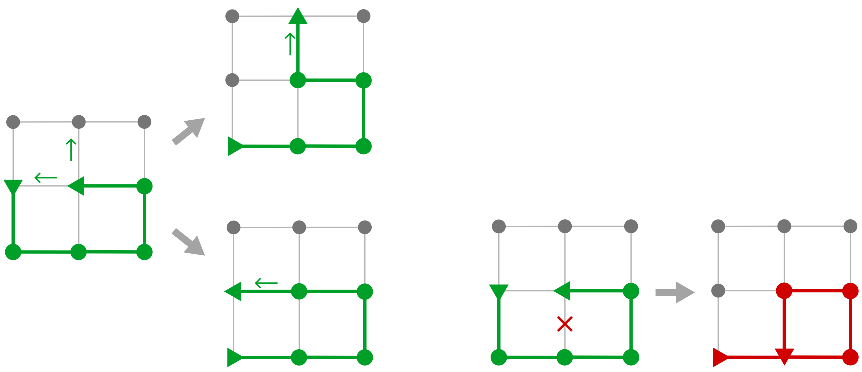
Then the next position of the snake becomes $S^{t+1} = (v, s_1^t, \dots, s_{\ell-1}^t)$. In other words, we add $v$ to the beginning of the path and remove $s_\ell^t$ from the end.

By our rules, $S^{t+1}$ should still form a simple path. It follows that the head can move to any adjacent vertex that is either unoccupied or the current tail vertex, as depicted in Figure 2.3a. More formally, the head must move to some vertex in $N(s_1^t) \cap \left(\overline{S^t} \cup \{s_\ell^t\}\right)$. In Figure 2.3b, we can see that if the snake moves to an occupied vertex that is not the tail, then the snake will no longer form a simple path. Hence, we forbid this type of movement.

An exception to the rules is made when the snake has length $\ell \leq 2$. For these shorter lengths, we do not allow the head to move to the tail vertex, since this would allow the snake to turn around. At the end of Section 2.4, we will see that this exception makes a significant difference.

## 2.3 Eating an apple

At the start of the game, the location of the first apple can be any vertex. Let $a$ be the first apple location, then the snake automatically starts on $a$ with $S^0 = (a)$. The next apple is placed on some vertex $a' \neq a$. The game then continues as follows. At each point in the game, there will be exactly one apple on the graph. The snake eats the apple by moving its head to this vertex, at which point a new apple is immediately placed on one of the unoccupied vertices. We will sometimes refer to the apple location as the apple itself. Let $S^t = (s_1^t, \dots, s_\ell^t)$ be the current snake position and $a$ the current apple location. Suppose the head moves from $s_1^t$ to $a$. Then the snake eats the apple on $a$ and the next snake position will be $S^{t+1} = (a, s_1^t, \dots, s_\ell^t)$. So unlike "normal" movement, $s_\ell^t$ is not removed from the path and the snake grows one vertex longer.



(a) The snake can move to an unoccupied vertex or the tail vertex.

(b) After an illegal move, the snake no longer forms a simple path.

Figure 2.3: The rules for the snake's movement are demonstrated on a grid graph.

**Observation 2.1.** *Let $a$ be the location of the apple at time $t$, then $a \notin S^t$.*

*Proof.* First, we note that $s_1^t \neq a$, since otherwise the apple would be eaten and no longer be on the graph.

Suppose $a = s_i^t$ for some $i \in \{2, \ldots, \ell\}$. Then the head must have been on $a$ at some earlier point after which $a$ has remained occupied by some part of the snake up until time $t$. But we know that the apple on $a$ could only have been placed at a time when $a$ was unoccupied. Hence, the head must have reached $a$ sometime after the apple was placed, which would imply that the apple was already eaten and is no longer on the graph. □

## 2.4 Winning and losing conditions

If the snake manages to reach length $|V|$, then we say the snake wins. Note that when this happens, the snake will occupy all the vertices. Hence, there are no more vertices an apple could be placed on.

If there is no vertex the snake can move to, then the snake loses. To be more precise, if $\ell < |V|$, the snake is in the position $S = (s_1, \ldots, s_\ell)$ and

$$N(s_1) \cap \left(\overline{S} \cup s_\ell\right) = \emptyset,$$

then the snake loses.

We want to avoid strategies where the snake can keep moving around without making progress. Imagine, for example, a scenario where the snake will lose if it eats the apple, but it can keep moving in cycles, postponing its loss forever. To this end, the snake will also lose if it repeats a previous position. More precisely, if for the current position $S^t$, there is some $t' < t$ such that $S^{t'} = S^t$, then the snake loses. Note that the snake can never repeat a position it was in at a shorter length. Furthermore, any strategy that repeats a position can be reformulated as one where the position is not repeated: we simply remove the set of moves between the two identical positions. Because of this, we will not be very careful with this rule when formulating winning snake strategies. If we find a winning strategy that violates it, we know there exists a winning strategy that does adhere to the rule.

Our aim is to determine on which graphs the snake can always win, regardless of the apple placement. To this end, we will often view it as a two-player game: one player controls the snake, and the other places the apples. While the *snake* tries to grow to length $|V|$, the *apple placer* tries to prevent the snake from doing so. We will usually assume both players play perfectly and always use a winning strategy if possible. When the snake has a winning strategy on a graph, we will call the graph *snake-winnable*. We will refer to the problem of determining whether a graph is snake-winnable as the *snake problem*.

To conclude our overview of the game, we make a few observations regarding snake-winnable graphs.

**Observation 2.2.** *If $G$ does not contain a Hamiltonian path, then $G$ is not snake-winnable.*

*Proof.* Since the snake must always form a simple path in $G$, if it reaches length $|V|$ it will form a Hamiltonian path. □

**Observation 2.3.** *If $G$ is Hamiltonian, then $G$ is snake-winnable.*

*Proof.* Since $G$ is Hamiltonian, there is some simple cycle $C$ in $G$ that contains all vertices. The snake can keep moving along this cycle. Since any apple will be placed on $C$, it will eventually be eaten by the snake. By following this strategy, the snake can keep growing until it covers the entire cycle, which contains all vertices. □

**Observation 2.4.** *Suppose $G = (V, E)$ has a spanning subgraph $G' = (V, E')$ that is snake-winnable, then $G$ is snake-winnable.*

*Proof.* Since $E' \subseteq E$, any move that is legal for the snake on $G'$ is also legal on $G$. It follows that if the snake has a winning strategy on $G'$, then it can use the same strategy to win on $G$. □

Note that Observation 2.4 also implies that if $G$ is not snake-winnable, then none of the spanning subgraphs of $G$ can be snake-winnable.

**Observation 2.5.** *For a graph $G = (V, E)$, if there is some $v \in V$ with degree $d(v) = 1$, then $G$ is not snake-winnable.*

*Proof.* Suppose there is some vertex $v \in V$ with $d(v) = 1$ and let $u$ be the only neighbor of $v$. Since $|V| \geq 3$, the first apple can be placed on some vertex that is not $v$. The second apple is then placed on $v$. When the snake eats the second apple, we must have $s_1 = v$ and $s_2 = u$. But since the snake has length 2, the head is not allowed to move to $u$, which means there is no vertex the head can move to. □

In the proof of Observation 2.5, we used that at length 2, the head cannot move to the tail vertex. In fact, without this restriction, the observation would not hold, as $P_3$, the path graph on three vertices, would be snake-winnable.

# 3. Snake on grid graphs

As we saw in Section 2, the original game of Snake can be described on a grid graph. In fact, the game is usually played on a subclass of grid graphs known as rectangular grid graphs. In this section, we will study the game on this subclass, as well as the more general class of grid graphs.

## 3.1 Rectangular grid graphs

In most versions of the game, the playing field of the snake is a rectangular area. This can be described as a *rectangular grid graph*, which has an embedding as a rectangular grid.

**Definition 3.1.** *A* ***rectangular grid graph*** $G = (V, E)$ *is a graph that has an embedding with* $V = [m] \times [n]$ *with* $m, n \in \mathbb{N}$*. For any* $u, v \in V$ *we have* $uv \in E$ *if and only if* $\|u - v\| = 1$*.*

When covering rectangular grid graphs, we will often refer to the embedding as the graph itself. In some cases, we will refer to a vertex by their coordinates in the embedding.

If either $m = 1$ or $n = 1$, then $G$ is a path graph and, thus, has a vertex of degree 1. By Observation 2.5, it follows that these graphs are not snake-winnable. We will show that all rectangular grid graphs that are not path graphs are snake-winnable. The following observation will be helpful.

**Observation 3.2.** *All rectangular grid graphs are bipartite.*

*Proof.* Let $G$ be a rectangular grid graph and $(x, y)$ be a vertex of $G$. The only possible neighbours of $(x, y)$ are $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, and $(x, y - 1)$. In other words, a vertex can only be adjacent to vertices with a coordinate sum that differs by exactly 1. It follows that we can partition the vertices based on the parity of their coordinate sum. □

Let us first turn our attention to rectangular grid graphs with an even number of vertices. These even-sized graphs are exactly those with at least one of $m$ or $n$ even. For $m, n \geq 2$, these graphs are known to be Hamiltonian [2]. For completion, we present our own proof by demonstrating how to construct such a Hamiltonian cycle.

**Lemma 3.3.** *Let* $G = (V, E)$ *be an* $m \times n$ *rectangular grid graph with* $m, n \geq 2$ *and at least one of* $m$ *or* $n$ *even. Then* $G$ *is Hamiltonian.*

*Proof.* Since an $m \times n$ rectangular grid graph is isomorphic to an $n \times m$ rectangular grid graph, without loss of generality, we may assume $m$ is even. Let $m = 2x$ for some $x \in \mathbb{N}$. Our Hamiltonian cycle will be constructed by linking several paths together. An example of this construction can be found in Figure 3.1.

For $i \in \{1, \ldots, x\}$, let $P_i$ be the path

$$P_i = ((2i - 1, 2), (2i - 1, 3), \ldots, (2i - 1, n), (2i, n), (2i, n - 1), \ldots, (2i, 2)) .$$

We can combine these paths to form a longer path $P = (P_1, P_2, \ldots, P_x)$. Then $P$ forms a simple path from $(1, 2)$ to $(2x, 2) = (m, 2)$. Let $P'$ be the path

$$P' = ((m, 1), (m - 1, 1), \ldots, (1, 1)) . \tag{1}$$

Note that $P'$ is completely disjoint from all the $P_i$, and therefore also disjoint from $P$. Furthermore, $P$ contains all vertices in $V$ that do not lie on $P'$. It follows by combining $P$ and $P'$, we obtain a Hamiltonian cycle. □

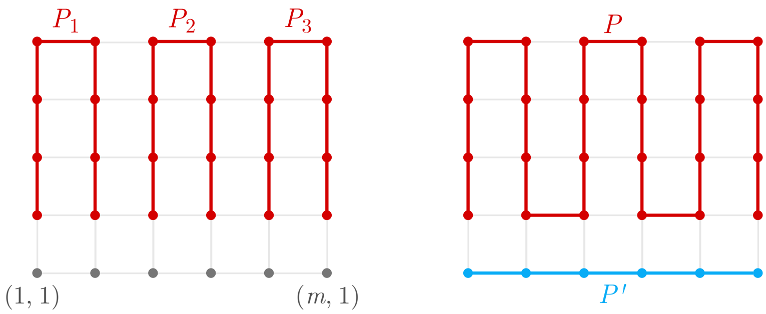



Figure 3.1: Constructing a Hamiltonian cycle on an even-sized rectangular grid graph.

**Corollary 3.4.** *Let $G = (V, E)$ be an $m \times n$ rectangular grid graph with $m, n \geq 2$ and at least one of $m$ or $n$ even. Then $G$ is snake-winnable*

*Proof.* By Observation 2.3, if a graph is Hamiltonian, then it is snake-winnable. □

Next, we will consider odd-sized rectangular grid graphs. Again, we only focus on rectangular grid graphs that are not path graphs. Unlike our previous result, these graphs turn out to be non-Hamiltonian.

**Observation 3.5.** *If $G = (V, E)$ is an odd-sized rectangular grid graph, then $G$ is non-Hamiltonian.*

*Proof.* By Observation 3.2, $G$ is bipartite. Bipartite graphs do not contain any odd cycles. For $G$ to be Hamiltonian, we would need a cycle of length $|V|$, but since $|V|$ is odd, this is not possible. □

Although odd-sized grid graphs are non-Hamiltonian, we can construct a cycle that contains all vertices except for one. In fact, we can find two of these cycles that differ by exactly one vertex. Thus, the union of these two cycles will contain all vertices. In Section 3.2.1, we will see that this type of two-cycle structure plays an important role in snake strategies on the more general class of grid graphs. Furthermore, we will introduce a generalization of this structure in Section 7.2, which is always snake-winnable.

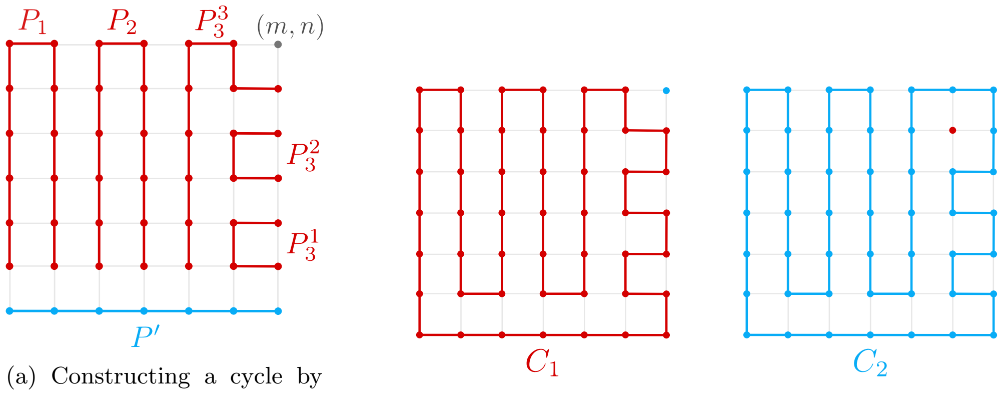
For odd-sized rectangular grid graphs, an example of the construction of these two cycles be found in Figure 3.2a and is done as follows.

**Lemma 3.6.** *Let $G = (V, E)$ be an $m \times n$ rectangular grid graph, with $m, n \geq 2$ and both odd. Then $G$ contains two different cycles $C_1$ and $C_2$, both of length $|V| - 1$, with the following property: there is a vertex $v$ on $C_1$ and a vertex $u$ on $C_2$ such that by replacing $v$ by $u$ on $C_1$, we obtain $C_2$.*

*Proof.* Let $m = 2x + 1$ and $n = 2y + 1$ for some $x, y \in \mathbb{N}$. We will first construct $C_1$, this is depicted in Figure 3.2a. The paths $P_1$ up to $P_{x-1}$ are constructed in the same manner as for the proof of Lemma 3.3. We construct the path $P_x$ by linking the following paths:

$$P_x^l = ((2x-1, 2), (2x-1, 3), \ldots, (2x-1, 2y+1), (2x, 2y+1), (2x, 2y), (2x+1, 2y))$$
$$P_x^j = ((2x+1, 2j+1), (2x, 2j+1), (2x, 2j), (2x+1, 2j))$$

for $j \in \{1, \ldots, y-1\}$. We combine these paths to $P_x = \left(P_x^y, \ldots, P_x^1\right)$, which we then add to the previous $P_i$ to form $P = (P_1, \ldots P_x)$. Combining $P$ with the path $P' = ((2x+1, 1), (2x, 1) \ldots, (1, 1))$ we obtain the cycle $C_1$ that contains all vertices except $(2x + 1, 2y + 1) = (m, n)$.




(a) Constructing a cycle by combining several paths.

(b) The two cycles $C_1$ and $C_2$ differ by one vertex.

Figure 3.2: Constructing two different cycles of length $|V| - 1$ on an odd-sized rectangular grid graph.

In $C_1$ we can exchange $(2x, 2y)$ for $(2x + 1, 2y + 1)$ to obtain a cycle of length $|V| - 1$ that contains all vertices except for $(2x, 2y)$. This gives us the second cycle $C_2$. The two cycles are depicted in Figure 3.2b. □

We will now use these two cycles to define a winning strategy for the snake on rectangular grid graphs with an odd number of vertices.

**Theorem 3.7.** *Let $G = (V, E)$ be an $m \times n$ rectangular grid graph with $m, n \geq 2$. Then $G$ is snake-winnable.*

*Proof.* By Corollary 3.4, we already know that if at least one of $m$ or $n$ is even, $G$ is snake-winnable. It remains to show that if both $m$ and $n$ are odd, then the snake still has a winning strategy.

Let the cycles $C_1$ and $C_2$ be as in Lemma 3.6. If the first apple appears on $(m-1, n-1)$, then the snake starts by moving clockwise along $C_1$. Similarly, if the first apple appears on $(m, n)$, the snake starts by moving clockwise along $C_2$. If the first apple appears on any other vertex, then the snake can pick either of the two cycles to move along. Note that while the snake has length at most $|V| - 2$, it can always decide to switch to the other cycle when it reaches $(m - 1, n)$. So until it reaches length $|V| - 1$, the snake always moves along one of the two cycles and switches whenever the apple is placed outside of its current cycle. By doing so, the snake can guarantee that once it eats the second to last apple and grows to length $|V| - 1$ it is positioned on one of the two cycles. Furthermore, it occupies all vertices on this cycle and thus the final apple will be placed on either $(m - 1, n - 1)$ or $(m, n)$. It can then repeatedly move to its tail until the head reaches $(m - 1, n)$, from which the snake can move to the vertex with the final apple. □

For Theorem 3.7, the snake uses a strategy where it can switch between two cycles. We will see this type of strategy return quite a few times, both on grid graphs as well as more general graph classes.

## 3.2 Grid graphs

Imagine that the playing field of the snake now includes obstacles that the snake has to avoid, as depicted in Figure 3.3. The snake no longer plays on a rectangular grid graph, as the playing field can now contain holes and have a non-rectangular outer boundary. To describe this game variant with obstacles, we turn to the more general class of *grid graphs*.

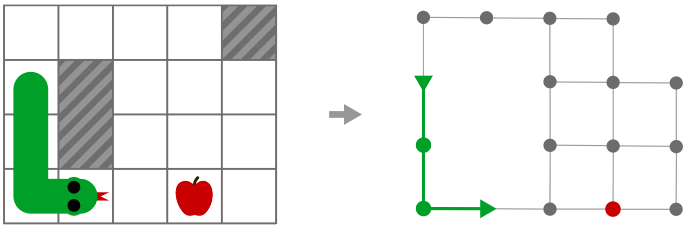


Figure 3.3: A grid graph can be used to describe the game of Snake with obstacles.

**Definition 3.8.** *A graph $G = (V, E)$ is a **grid graph** if and only if it is a vertex-induced subgraph of some rectangular grid graph.*

Note that this also allows grid graphs to have "holes": points inside the outer boundary that are not vertices. There is a subclass known as *solid grid graphs*, which can have a non-rectangular shape, but do not contain any holes. We will take a closer look at this subclass in Section 3.3.

Since a grid graph is a vertex-induced subgraph of some rectangular grid graph, we maintain that for every pair of vertices $u, v \in V$, $uv \in E$ if and only if $\|u - v\| = 1$. We also maintain the following

**Observation 3.9.** *Any grid graph is bipartite.*

*Proof.* Any grid graph is an induced subgraph of a rectangular grid graph. By Observation 3.2 rectangular grid graphs are bipartite. It follows that grid graphs are also bipartite. □

### 3.2.1 Odd-sized grid graphs

From Observation 3.9, we obtain that odd-sized grid graphs are non-Hamiltonian. We will see that for such a graph to be snake-winnable, we need a similar structure to the one we found for Theorem 3.7.

**Definition 3.10.** *The **theta graph** $\Theta(p, q, r)$ is constructed by taking two vertices $u$ and $v$ and connecting them by three internally disjoint paths of lengths $p$, $q$ and $r$. If at least one of $p$, $q$ or $r$ is 0, then $u = v$.*

For Theorem 3.7, we used the two cycles $C_1$ and $C_2$. By combining these two cycles, we obtain the graph $\Theta(|V| - 3, 2, 2)$, as depicted in Figure 3.4. It turns out that for odd-sized grid graphs, the snake-winnability is solely determined by the existence of $\Theta(|V| - 3, 2, 2)$ as a spanning subgraph. To prove this, we first turn to the more general class of bipartite graphs.

**Observation 3.11.** *Let $G = (X \cup Y, E)$ be a bipartite graph. If $||X| - |Y|| > 1$, then $G$ is not snake-winnable.*

*Proof.* Any path in $G$ must alternate between $X$ and $Y$. Hence, $G$ can only contain a Hamiltonian path if $|X| = |Y|$ or $||X| - |Y|| = 1$. By Observation 2.2, it follows that if $||X| - |Y|| > 1$, then $G$ is not snake-winnable. □

**Theorem 3.12.** *Let $G = (V, E)$ be an odd-sized bipartite graph with partition $V = X \cup Y$. Then $G$ is snake-winnable if and only if $\Theta(|V| - 3, 2, 2)$ is a spanning subgraph of $G$.*

*Proof.* If $\Theta(|V| - 3, 2, 2)$ is a spanning subgraph of $G$, then the snake can use the strategy from Theorem 3.7 to win.

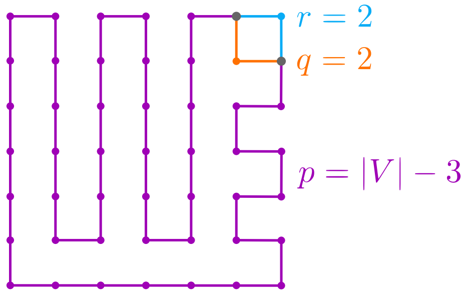



Figure 3.4: Odd-sized rectangular grid graphs have a $\Theta(p, q, r)$ spanning subgraph with $p = |V| - 3$, $q = 2$ and $r = 2$.

It remains to show that if there is no $\Theta(|V| - 3, 2, 2)$ spanning subgraph, then the apple placer has a winning strategy. We will approach this as follows. As the snake reaches length $|V| - 1$, there are at most two vertices the snake can move to: the only remaining unoccupied vertex and the tail vertex. If the head and the unoccupied vertex are in the same part, then the head can only move to the tail. By cleverly placing the apples, we will show that the apple placer can guarantee this is the case. Furthermore, due to the previous apple placement, the snake will create a $\Theta(|V| - 3, 2, 2)$ spanning subgraph if it moves to the tail. Hence, the snake cannot move to the unoccupied vertex nor the tail vertex, and will thus lose.

Assume $\Theta(|V| - 3, 2, 2)$ is not a spanning subgraph of $G$. Since $G$ is odd-sized, by Observation 3.11 we know $G$ can only be snake-winnable if $||X| - |Y|| = 1$. Without loss of generality, we will assume that $|X| = |Y| + 1$.

Consider the moment the snake reaches length $|V| - 3$. Since $|V| - 3$ is even, the snake occupies the same number of vertices of $X$ as of $Y$. Hence, there is one vertex $y \in Y$ that is unoccupied. The apple placer places the next apple on $y$. When the snake eats the apple on $y$, it reaches length $|V| - 2$, which is an odd number. Since the snake alternates between $X$ and $Y$, both its head and tail must be in $Y$ as depicted in Figure 3.5. Hence, the two remaining unoccupied vertices are both in $X$.

Suppose both unoccupied vertices are adjacent to the head and the tail. Then between the head and tail of the snake, we have the path of length $|V| - 3$ formed by the snake itself, and two paths of length 2, each through one of the unoccupied vertices. Thus, we obtain a $\Theta(|V| - 3, 2, 2)$ spanning subgraph of $G$. Since we assumed such a spanning subgraph did not exist, we can conclude that at least one of the unoccupied vertices is not adjacent to

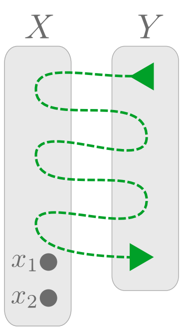
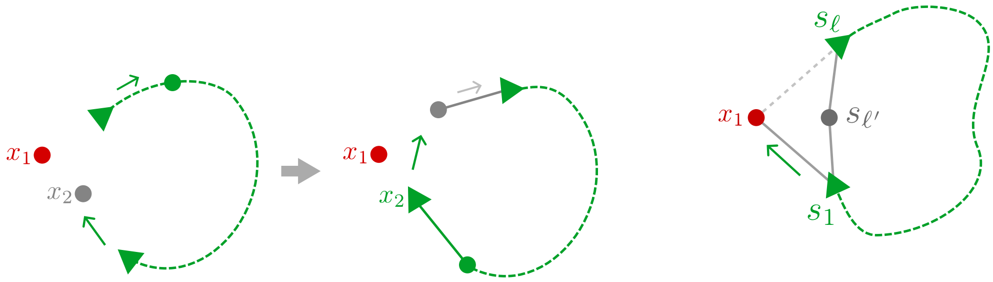



Figure 3.5: The snake always alternates between $X$ and $Y$. When the snake reaches length $|V| - 2$ with its head in $Y$, the two remaining unoccupied vertices are in $X$.




(a) The snake has to move to $x_2$ first. From $x_2$, the snake can only move to the previous tail.

(b) The position of the snake before eating the apple on $x_1$.

Figure 3.6: After eating the apple on $x_1$, the snake can only move to $s_l$, but this move requires $\Theta(|V|-3,2,2)$ spanning subgraph.

both the head and the tail. Let this be $x_1$ and let $x_2$ be the other unoccupied vertex. The apple placer places the next apple on $x_1$.

Suppose the snake immediately eats $x_1$, without moving to some other vertex first. Then the tail would remain in the same place. Furthermore, this requires $x_1$ to be adjacent to the head, which means it is not adjacent to the tail. Hence, from $x_1$ the snake can only move to another unoccupied vertex. But since $x_2$ is the only remaining unoccupied vertex and $x_1$ and $x_2$ are both in $X$, this is not possible. It follows that the snake must move to some other vertex first before eating the apple on $x_1$. At length $|V|-2$, the snake cannot move to its tail since this would create an odd cycle. Thus, the only move the snake can make is to $x_2$.

After moving to $x_2$, the previous tail vertex becomes unoccupied, as depicted in Figure 3.6a. From $x_2$, the snake cannot move to $x_1$ or the current tail vertex. Hence, it has to move to the previous tail vertex. Continuing this reasoning, we find that the only thing the snake can do until it eats the apple is repeatedly moving to the previous tail vertex. Of course, this type of move requires that the previous tail vertex is adjacent to the current head. Since the previous tail vertex is also adjacent to the current tail, the snake must be moving along a cycle of length $|V|-1$, consisting of the snake itself and the previous tail vertex.

Let $S$ be the position of the snake right before it moves to $x_1$, which is depicted in Figure 3.6b. Let $s_1$ be the head vertex and $s_\ell$ the tail vertex. Furthermore, let $s_{\ell'}$ be the previous tail vertex, that is adjacent to both $s_1$ and $s_\ell$. Note that since the head moves from $s_1$ to $x_1$, $s_1$ and $s_\ell$ must both be in $Y$, and $s_{\ell'}$ in $X$. After eating the apple on $x_1$, $s_\ell$ remains the tail vertex. From $x_1$, the snake cannot move to $s_{\ell'}$, since they are both in $X$. If the snake can move to $s_\ell$, then the paths $(s_1, x_1, s_\ell)$ and $(s_1, s_{\ell'}, s_\ell)$, together with $S$ form a $\Theta(|V|-3,2,2)$ spanning subgraph of $G$. Since we assumed such a spanning subgraph did not exist, we can conclude that there is no vertex the snake can move to form $x_1$, and will thus lose. □

**Corollary 3.13.** *Let $G=(V,E)$ be an odd-sized grid graph. Then $G$ is snake-winnable if and only if $\Theta(|V|-3,2,2)$ is a spanning subgraph of $G$.*

*Proof.* Every odd-sized grid graph is bipartite. □

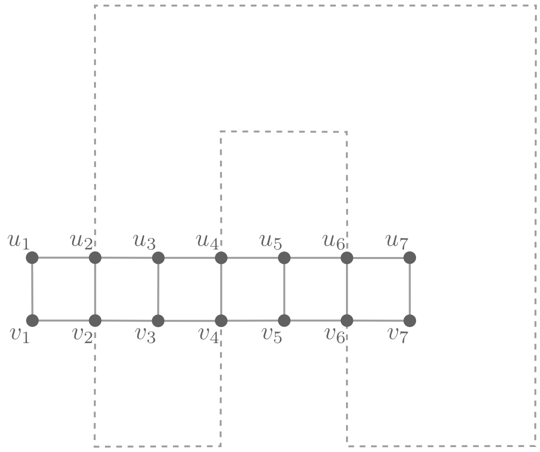



Figure 3.7: A even-sized non-Hamiltonian grid graph that is snake-winnable. The dashed lines indicate paths, which should be made sufficiently long to be embedded as depicted.

### 3.2.2 Even-sized grid graphs

Corollary 3.13 gives us a characterization of all odd-sized grid graphs that are snake-winnable. Ideally, we would find a similar characterization for even-sized grid graphs. Since we already know Hamiltonian grid graphs are snake-winnable, the question becomes whether there are any even-sized non-Hamiltonian grid graphs that are snake-winnable. The graph in Figure 3.7 shows us that these types of graphs do exist. In Figure 3.8, we see this graph contains a cycle of length $|V| - 2$. Since grid graphs cannot contain cycles of odd length, $|V| - 2$ must be even, and thus $|V|$ is even. We first show that this graph is indeed non-Hamiltonian.

**Observation 3.14.** *The graph in Figure 3.7 is non-Hamiltonian.*

*Proof.* For any vertex with degree 2, both incident edges must be included in any Hamiltonian cycle. It follows that if the graph in Figure 3.7 would have a Hamiltonian cycle, then the dashed paths, as well as $(u_2, u_1, v_1, v_2)$ and $(u_6, u_7, v_7, v_6)$ must be part of it. Combined, this gives us a path from $u_4$ to $v_4$ that contains all vertices except for $u_3$, $v_3$, $u_5$, and $v_5$. To form a Hamiltonian cycle, we must find a $u_4v_4$-path with exactly these remaining four as internal vertices. But this is not possible, and thus the graph in Figure 3.7 is not Hamiltonian. □

To prove the snake has a winning strategy on this graph, we first make the following observation.

**Observation 3.15.** *Suppose the snake has length $|V|-2$ and is positioned on some cycle $C$. Furthermore, suppose the snake covers all vertices on $C$, and the two remaining unoccupied vertices are adjacent to each other. If both unoccupied vertices are adjacent to some vertex in $C$, then the snake will win.*

*Proof.* Let $u$ and $v$ be the two unoccupied vertices and suppose they are adjacent to the vertices $u'$ and $v'$ on $C$, respectively. If the next apple is on $u$, then the snake moves along $C$ until it reaches $u'$. From there, it moves to $u$. It can then move to $v$ to eat the final apple. Similarly, if the next apple is on $v$ then the snake moves along $C$ until it reaches $v'$. From there, it moves to $v$, and then $u$ to eat the final apple. □

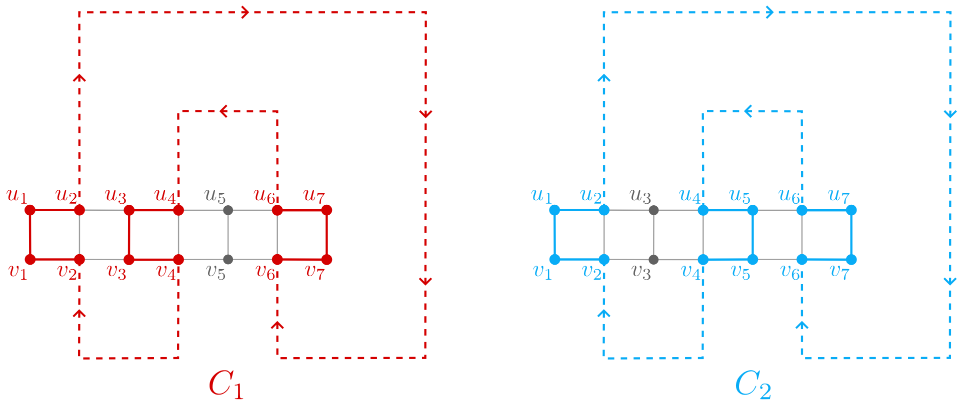



Figure 3.8: The cycles $C_1$ and $C_2$, both of length $|V| - 2$.

**Lemma 3.16.** *There are non-Hamiltonian grid graphs with an even number of vertices that are snake-winnable.*

*Proof.* Let $G = (V, E)$ be a graph as depicted in Figure 3.7. In Figure 3.8 we can see the cycles $C_1$ and $C_2$ of $G$, both have length $|V| - 2$. The strategy of the snake starts similarly to the strategy from Theorem 3.7. The snake moves along $C_1$ or $C_2$ and changes to the other cycle if the apple is placed outside of its current cycle. Note that while the snake has length at most $|V| - 4$, it can choose which cycle to move along each time it reaches $u_4$. By employing this strategy, the snake will be positioned on one of the two cycles when it reaches length $|V| - 3$. By symmetry, we may assume this is $C_1$.

If the next apple is on $C_1$, then the snake can eat the apple by moving along $C_1$. After doing so, it covers the entire cycle, and the two remaining unoccupied vertices are $u_5$ and $v_5$. By Observation 3.15, the snake will win.

If the next apple is not on $C_1$, then it is either on $u_5$ or $v_5$. If it is on $v_5$, then the snake moves along $C_1$ until it reaches $u_4$. It then first moves to $u_5$, and then to $v_5$ to eat the apple. By doing so, the snake will be positioned on $C_2$, with $u_3$ and $v_3$ the remaining unoccupied vertices. By Observation 3.15, the snake will win.

If the next apple is on $u_5$, then the snake moves along $C_1$ until it reaches $v_6$. It then first moves to $v_5$, and then to $u_5$ to eat the apple. The snake will now be positioned on a new cycle, depicted in Figure 3.9. The two remaining vertices are $u_7$ and $v_7$. By Observation 3.15, the snake will win. □

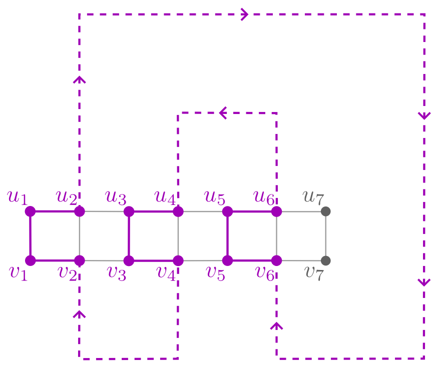



Figure 3.9: If the apple appears on $u_5$, the snake moves to a new cycle.

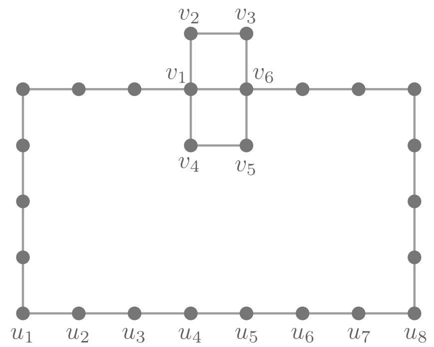



Figure 3.10: A grid graph with a $\Theta(|V|-5,3,3)$ spanning subgraph that is not snake-winnable.

Interestingly, the graph in Figure 3.7 has a $\Theta(|V|-5,3,3)$ spanning subgraph. In fact, it has multiple $\Theta(|V|-5,3,3)$ spanning subgraphs. However, it turns out having such a spanning subgraph is not sufficient for a graph to be snake-winnable. In Figure 3.10 we see a graph that has a $\Theta(|V|-5,3,3)$ spanning subgraph but is not snake-winnable. This spanning subgraph can be formed with the paths $(v_1, v_2, v_3, v_6)$, $(v_1, v_4, v_5, v_6)$ and the lower $v_1v_6$-path, through $(u_1, \ldots, u_8)$.

**Lemma 3.17.** *There are grid graphs with a* $\Theta(|V|-5,3,3)$ *spanning subgraph that are not snake-winnable.*

*Proof.* Let $G = (V, E)$ be the graph in Figure 3.10. When the snake reaches length 6, one of the vertices in $\{u_1, \ldots, u_8\}$ must be unoccupied. The apple placer places the next apple on this vertex. After eating the apple and growing to length 7, the vertices in $\{v_1, \ldots, v_6\}$ are too far away from the head to be on the snake. Hence, none of the vertices in $\{v_1, \ldots, v_6\}$ are occupied.

Suppose the snake now wants to occupy both pairs $\{v_2, v_3\}$ and $\{v_4, v_5\}$. Then it must move in a way similar to Figure 3.11. But as we can see, this will cause the snake to lose. It follows that while the snake has length at least 7 and at most $|V|-4$, one of the pairs $\{v_2, v_3\}$ and $\{v_4, v_5\}$ will be unoccupied. At these lengths, the snake can also no longer turn around: if it is currently moving clockwise, it cannot change to moving counterclockwise. By symmetry, we may assume the snake is moving clockwise.

When the snake reaches length $|V|-4$, one the pairs $\{v_2, v_3\}$ and $\{v_4, v_5\}$ will be unoccupied. By symmetry, we may assume $v_2$ and $v_3$ are unoccupied. The apple placer places

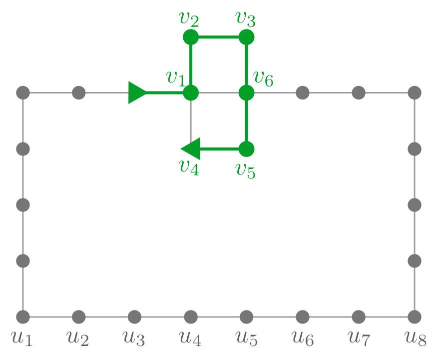
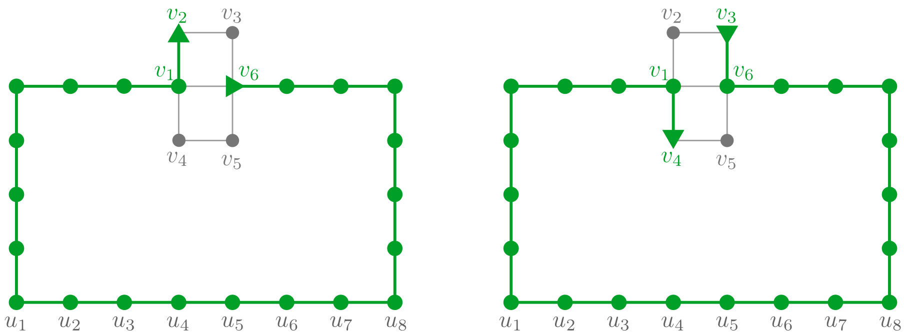



Figure 3.11: The type of maneuver the snake has to make to occupy $\{v_1, \ldots, v_6\}$ at length 7 or longer.




(a) The position of the snake as it reaches length $|V| - 3$.

(b) The position of the snake as it reaches length $|V| - 2$.

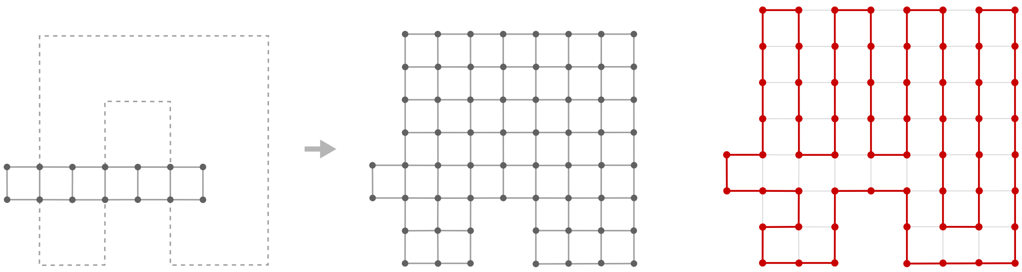
Figure 3.12: When the snake reaches length $|V| - 2$, the next apple can be placed on $v_5$ from where the snake will be stuck.

the next apple on $v_2$. Since we assumed the snake is moving clockwise, the snake can only eat the apple while in the position depicted in Figure 3.12a. The apple placer places the next apple on $v_4$. When the snake eats the apple on $v_4$, it must be in the position depicted in Figure 3.12b, with its tail on $v_3$. The apple placer places the next apple on $v_5$. Since $v_5$ is the only vertex the head can move to, the snake is forced to eat the apple on its next move. So the head moves to $v_5$, but the tail remains on $v_3$. From $v_5$, there is no vertex the snake can move to, and thus it will lose. □

While having a $\Theta(|V|-5, 3, 3)$ spanning subgraph is not a sufficient condition for a graph to be snake-winnable, the question of whether it is necessary remains open. Answering this question, however, would still not provide a characterization of all even-sized non-Hamiltonian grid graphs that are snake-winnable. Our winnable example in Figure 3.7 had multiple $\Theta(|V| - 5, 3, 3)$ spanning subgraphs. For future research, one might look at more complicated structures consisting of multiple $\Theta(|V| - 5, 3, 3)$ spanning subgraphs to search for a characterization.

## 3.3 Solid grid graphs

Unlike grid graphs, *solid grid graphs* cannot contain holes. Neither the graph in Figure 3.7, nor the graph in Figure 3.10 are solid grid graphs. This leaves us to wonder what the



(a) Transforming the graph from Figure 3.7 into a solid grid graph.

(b) The resulting graph has a Hamiltonian cycle.

Figure 3.13: If we transform the graph from Figure 3.7 into a solid grid graph by adding vertices, then the resulting graph is Hamiltonian.

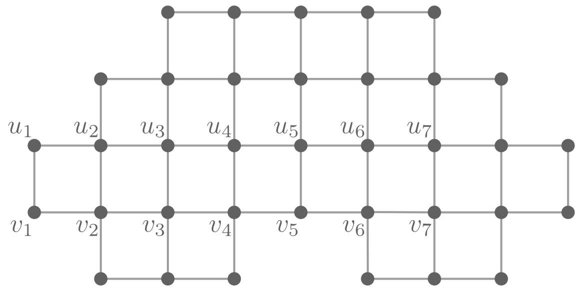



Figure 3.14: An even-sized solid grid graph that is non-Hamiltonian and snake-winnable.

snake-winnable solid grid graphs look like.

**Definition 3.18.** *A **solid grid graph** $G = (V, E)$ is a grid graph where every point within the outer boundary is a vertex of $G$.*

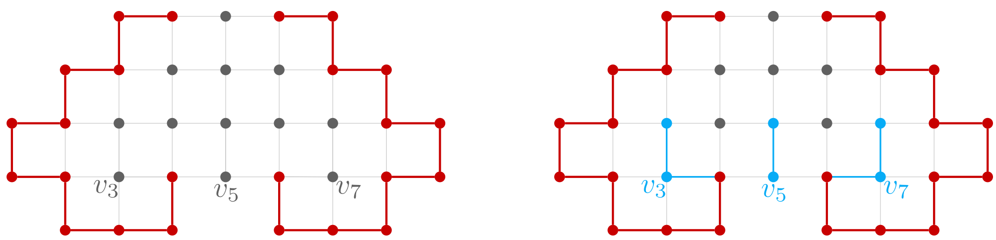
For odd-sized solid grid graphs, Corollary 3.13 holds, so we have a complete characterization of the snake-winnable graphs. For even-sized solid grid graphs, we would like to know if there are any that are non-Hamiltonian and snake-winnable. The example of a non-Hamiltonian even-sized graph from Figure 3.7 is not a solid grid graph. In Figure 3.13a we can see one way to transform it into a solid grid graph by adding vertices. But this new graph is Hamiltonian, as demonstrated in Figure 3.13b. However, there are even-sized solid grid graphs that are non-Hamiltonian and snake-winnable. One of these graphs is depicted in Figure 3.14.

**Observation 3.19.** *The graph in Figure 3.14 is non-Hamiltonian.*

*Proof.* For all vertices with degree 2, both incident edges will be part of any Hamiltonian cycle. This is depicted in Figure 3.15a. After adding these edges, both $v_3$ and $v_7$ have only two incident edges that can still be part of the cycle, as depicted in Figure 3.15b. But this leaves $v_5$ with only one incident edge for the cycle, from which it follows that we cannot form a Hamiltonian cycle. □

**Lemma 3.20.** *There are non-Hamiltonian solid grid graphs with an even number of vertices that are snake-winnable.*

*Proof.* Let $G = (V, E)$ be the graph in Figure 3.14 and let $C_1$ and $C_2$ be as depicted in Figure 3.16. Note that these two cycles have the same structure as the cycles in Figure 3.8. Therefore, we can employ the exact same strategy as used for Lemma 3.16. □




(a) All edges incident with a vertex of degree 2 must be part of any Hamiltonian cycle.

(b) The vertices $v_3$ and $v_7$ can only be added in one way, leaving $v_5$ with only one incident edge for the cycle.

Figure 3.15: The graph in Figure 3.14 is non-Hamiltonian.

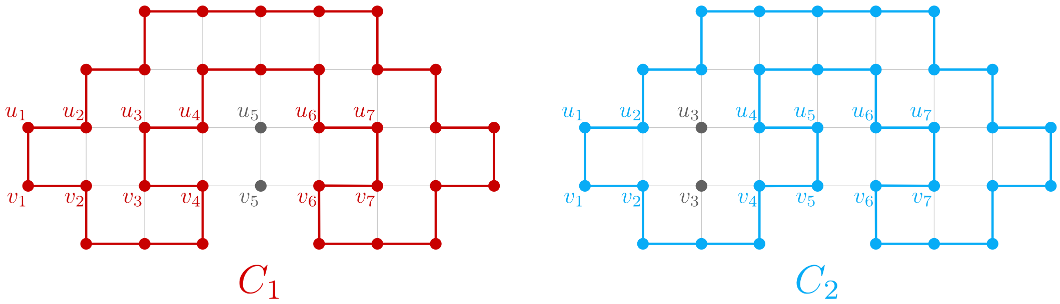



Figure 3.16: The cycles $C_1$ and $C_2$ have the same structure as the cycles in Figure 3.8.

While the graph in Figure 3.14 does have a $\Theta(|V| - 5, 3, 3)$ spanning subgraph, we do not know if this is a sufficient or even necessary condition for an even-size solid grid graph that is non-Hamiltonian to be snake-winnable.

# 4. The complexity of the snake problem

The complexity of a problem gives us an idea of how difficult it is to solve. First, we provide a brief overview of the complexity class NP, NP-hardness, and NP-completeness. For a more in-depth source on the topic, we refer to the book *Computational Complexity* by Papadimitriou [10].

A *decision problem* is a problem that can be answered with either "yes" or "no". For decision problems on graphs, we usually want to determine whether a graph has a certain property or not. In our case, this is whether or not the graph is snake-winnable.

A decision problem is in the complexity class NP if every "yes" instance can be verified in polynomial time. This is done with the use of a *certificate*, which, when given alongside the instance, allows us to verify in polynomial time that it is indeed a "yes" instance. For example, for the Hamiltonian cycle problem, if we are given a graph $G = (V, E)$ and a cycle containing all the vertices of $V$, then we can verify that this cycle indeed exists in $G$. The complexity class P consists of all decision problems that can be solved in polynomial time. For any of the problems in P, we can give the polynomial-time algorithm as a certificate, and thus $\mathsf{P} \subseteq \mathsf{NP}$. Quite famously, it has been conjectured that $\mathsf{P} \neq \mathsf{NP}$, however, this has not yet been proven.

A *polynomial-time reduction* is a transformation from an instance of problem $A$ to an instance of problem $B$, that can be done in polynomial time and has the following property. If the instance of $A$ is a "yes" instance, then it is always transformed into a "yes" instance of $B$. Conversely, if it is a "no" instance of $A$, then it is always transformed into a "no" instance of $B$. Suppose we have some algorithm to solve $B$. Then we can use it to solve $A$ by first using the transformation. This means that if we know $A$ is hard to solve, $B$ must be hard to solve as well.

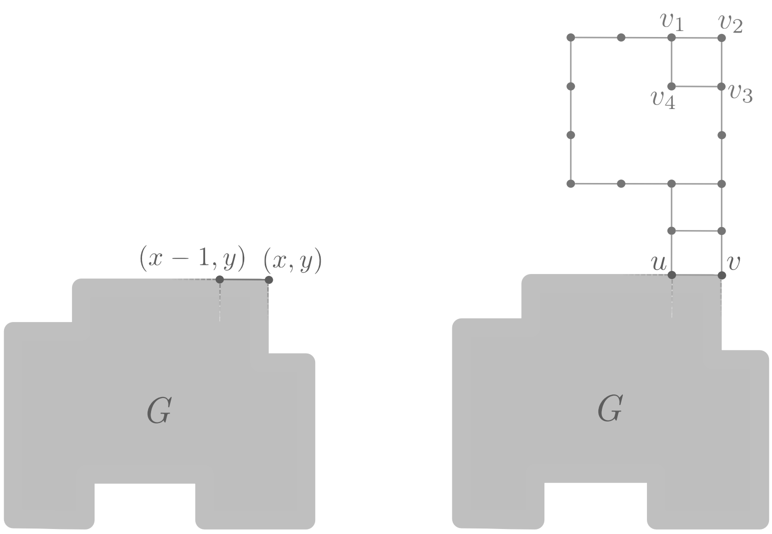
A decision problem $A$ is NP-*hard* if, for every problem in NP, there is a polynomial-time reduction to $A$. Informally, $A$ is at least as hard as every problem in NP. If a decision problem is NP-hard and also in NP itself, then it is NP-*complete*.

Recall that by Corollary 3.13, an odd-sized grid graph is snake-winnable if and only if it has a $\Theta(|V| - 3, 2, 2)$ spanning subgraph. We will use this characterization to show that the snake problem on grid graphs is NP-hard. To do so, we reduce from the Hamiltonian cycle problem on grid graphs, which is NP-complete [5]. By adding a gadget, we will create an odd-sized grid graph where the existence of an $\Theta(|V| - 3, 2, 2)$ as a spanning subgraph depends on the Hamiltonicity of the original graph.

**Theorem 4.1.** *The snake problem is* NP*-hard, even when restricted to grid graphs.*

*Proof.* Let $G = (V, E)$ be an instance of the Hamiltonian cycle problem on grid graphs. Since odd-sized grid graphs are never Hamiltonian, we may assume $G$ is even-sized. Let $v = (x, y)$ be the vertex on the top row of $G$ that is furthest to the right, as depicted in Figure 4.1a. This means the points $(x, y + 1)$ and $(x + 1, y)$ are not vertices of $G$. If $(x - 1, y)$ is also not a vertex of $G$, then $(x, y)$ has degree at most 1, which means $G$ cannot be Hamiltonian. Hence we may assume that $(x - 1, y)$ is also a vertex of $G$, and we denote $u = (x - 1, y)$. Since the degree of $v$ is at most 2, the edge $uv$ must be part of any Hamiltonian cycle.

For our reduction, we create a new graph $G' = (V', E')$ by attaching a gadget to $u$ and $v$, as depicted in Figure 4.1b. Since the gadget is odd-sized and we assumed $G$ was even-sized, $G'$ is an odd-sized grid graph. In $G'$, let the vertices $v_1, v_2, v_3, v_4$ be as depicted




(a) The two rightmost points of the top row of $G$.

(b) Attaching the gadget to $G$ to create $G'$.

Figure 4.1: The reduction from a grid graph $G$ to $G'$ by using a gadget.

in Figure 4.1b. Note that there is no cycle in $G'$ of length $|V'| - 1$ that contains both $v_2$ and $v_4$. Thus, any $\Theta(|V| - 3, 2, 2)$ spanning subgraph of $G'$ must consist the following three paths: $(v_1, v_2, v_3)$, $(v_1, v_4, v_3)$, and some $v_1v_3$-path that contains all vertices in $V' \backslash \{v_2, v_4\}$. Since the gadget only connects to $G$ at $u$ and $v$, to form the latter path $G$ must have a Hamiltonian path from $u$ to $v$. But since the edge between $uv$ is part of any Hamiltonian cycle, this Hamiltonian path from $u$ to $v$ exists if and only if $G$ is Hamiltonian. It follows that $G'$ has a $\Theta(|V| - 3, 2, 2)$ spanning subgraph if and only if $G$ is Hamiltonian. By Corollary 3.13, we obtain that $G'$ is snake-winnable if and only if $G$ is Hamiltonian, completing our reduction and proving the NP-hardness of the snake problem on grid graphs. □

Since all grid graphs are planar and bipartite, by Theorem 4.1 the snake problem is also NP-hard when restricted to either of these graph classes. For odd-sized bipartite graphs, Theorem 4.1 gives us the following.

**Corollary 4.2.** *The snake problem on odd-sized bipartite graphs is* NP*-complete.*

*Proof.* By Theorem 3.12, an odd-sized bipartite graph is snake-winnable if and only if it has a $\Theta(|V| - 3, 2, 2)$ spanning subgraph. Thus, we can give a $\Theta(|V| - 3, 2, 2)$ spanning subgraph as a certificate. □

Naturally, we would like to know whether we can also find a certificate for other graph classes. For Hamiltonian graphs, this can be the Hamiltonian cycle. Naively, we might consider giving the winning snake strategy as a certificate for all other snake-winnable graphs. However, it is unclear whether such a strategy can be formulated in polynomial size. We could, for example, try to describe the path the head has to take for each possible apple placement. But when the snake has length $\ell$, there are still $|V| - \ell$ possible locations for the next apple. Hence, the number of paths in such a strategy could be on the order of $|V|!$. Furthermore, these paths do not need to be simple and are therefore not easily bounded. We do know the snake cannot repeat a previous position, but especially at longer lengths the number of possible positions for the snake could be very large. The question of whether there is always a winning strategy that can be formulated in polynomial size remains open.

Unlike on grid graphs, the Hamiltonian problem on solid grid graphs does have a polynomial-time algorithm [11]. Hence, a reduction similar to the one from Theorem 4.1 would not work to show the snake problem is NP-hard on solid grid graphs. The question of whether the snake problem is also NP-hard on solid grid graphs remains open. Interestingly, the Nibbler food collection problem studied by De Biasi and Ophelders is similar to our snake problem and was found to be NP-hard on solid grid graphs[3]. In this game, the snake starts at a given length in a given position, and the location of each apple is known. These apples are all placed on the graph at the start of the game and no new apples appear when the snake eats an apple. The snake moves by the same rules as for our Snake game and the question is whether there is a path the snake can take that allows it to eat all the apples.

# 5. The girth of snake-winnable graphs

So far, we have seen a few strategies where the snake can switch between two cycles. If the snake wants to quickly switch between the cycles, then they must share many vertices. However, this also means the part where the cycles differ forms a smaller cycle. So to be able to execute such a strategy, the shortest cycle in the graph cannot be too large.

**Definition 5.1.** *The* ***girth*** *of a graph $G$, denoted $g(G)$, is the length of the shortest cycle in $G$.*

Grid graphs, for example, always have a girth of at least 4, since they do not contain any triangles. By Observation 2.3, any Hamiltonian graph is snake-winnable. Since any cycle graph is Hamiltonian, we can create snake-winnable graphs of arbitrarily large girth.

By Lemma 3.12, an odd-sized bipartite graph is snake-winnable if and only if it has a $\Theta(|V|-3,2,2)$ as a spanning subgraph. These winnable graphs always have girth 4, as $\Theta(|V|-3,2,2)$ has a cycle of length 4 and bipartite graphs cannot have any cycles of length 3. This raises the question of what can be said about the girth of non-Hamiltonian snake-winnable graphs. In Section 5.1, we will show that if the girth of a non-Hamiltonian graph is greater than 6, then it is not snake-winnable. In Section 5.2 we provide an example of a non-Hamiltonian snake-winnable graph that has a girth of 6, showing that this bound is tight.

## 5.1 Bounding the girth of non-Hamiltonian snake-winnable graphs

Let $G = (V, E)$ be a graph and $C$ be some cycle in $G$. We will say a cycle $C$ *contains* the snake if the entire path formed by the snake lies on $C$. To prove that any non-Hamiltonian graph with a girth greater than 6 is not snake-winnable, we will do the following. When the snake reaches length $|V|-3$, there are four possible scenarios: the snake can be contained in a cycle of length $|V|-3$, a cycle of length $|V|-2$, a cycle of length $|V|-1$, or in no cycle. Note that there can be no cycle of length $|V|$ that contains the snake since that would make the graph Hamiltonian. We will show that for each of these scenarios, the apple placer has a winning strategy. To do so, we first prove several lemmas that show that, as the snake grows longer, the girth required to make certain types of moves becomes smaller. In other words, if the girth is large, then as the snake grows long, its movement will become more limited.

**Lemma 5.2.** *Let $C$ be a cycle in $G$ that contains the snake and let $\ell$ be the current length of the snake. Suppose the head of the snake leaves $C$ and returns to $C$ after visiting $m$ vertices in $\overline{C}$. Then $g(G) \leq |C| - \ell + 2m + 2$.*

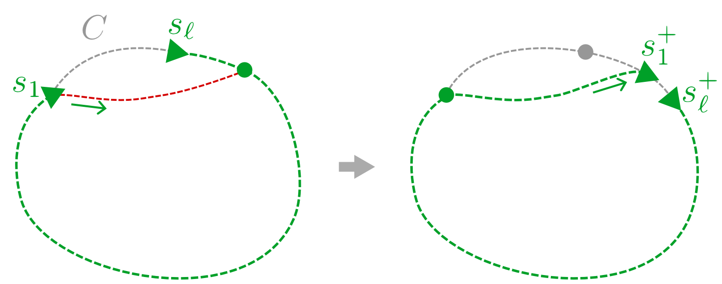
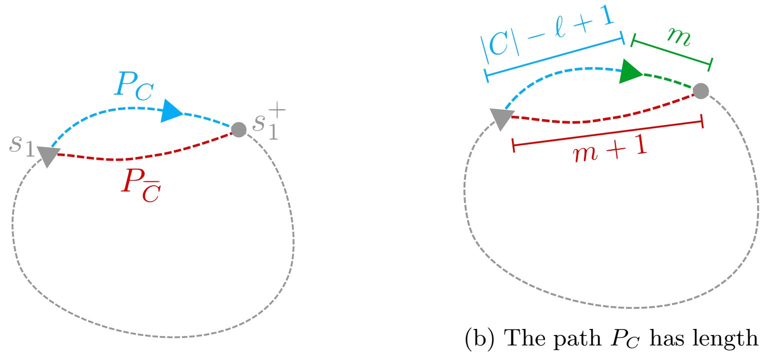



Figure 5.1: The snake leaves $C$ from $s_1$ and returns to $C$ at $s_1^+$.




(a) The paths $P_C$ on $C$ and $P_{\overline{C}}$ through $\overline{C}$.

(b) The path $P_C$ has length at most $|C| - \ell + 1 + m$ and $P_{\overline{C}}$ has length $m + 1$.

Figure 5.2: If the snake returns to $C$ after visiting $m$ vertices, $G$ must contain a cycle of length $|C| - \ell + 2m + 2$.

*Proof.* The idea behind the proof is as follows. The path the snake takes outside of $C$, together with the section of $C$ it "skips", will form a cycle. But the snake cannot just return anywhere on $C$, since it must move to an unoccupied vertex or the tail. Hence, the length of the section it can skip will be bounded. We will show that the length of this skipped section is at most $|C| - \ell + m + 1$. Combined with the path the snake takes through $\overline{C}$, this will result in a cycle of length at most $|C| - \ell + 2m + 2$.

We first note that if $\ell \leq 2m + 2$, then $|C| - \ell + 2m + 2 \geq |C|$. Since the existence of $C$ implies $g(G) \leq |C|$, the statement is trivially true. Hence, we may assume that $\ell > 2m + 2 > m$, which means that while the head moves through $\overline{C}$, the tail remains on $C$.

Suppose the head of the snake leaves $C$ and returns to $C$ after visiting $m$ vertices in $\overline{C}$, as depicted in Figure 5.1. Let $S = (s_1, \ldots, s_\ell)$ be the position of the snake right before the head leaves $C$. Similarly, let $S^- = (s_1^-, \ldots, s_\ell^-)$ be the position of the snake right before the head returns to $C$ and $S^+ = (s_1^+, \ldots, s_\ell^+)$ the position right after. By $P_C$ we denote the section of $C$ that the snake skips, as depicted in Figure 5.2a. More precisely, $P_C$ is the path the head would have taken had it stayed on $C$, with endpoints $s_1$ and $s_1^+$. When the snake is in position $S^- = (s_1^-, \ldots, s_\ell^-)$, all the unoccupied vertices on $C$ are between $s_1$ (from where the snake left $C$) and $s_\ell^-$. From $s_1^-$, the head will move to one of these unoccupied vertices or to $s_\ell^-$. Hence, the length of $P_C$ is maximized if the head returns to $C$ by moving to its tail and $s_1^+ = s_\ell^-$. This is depicted in Figure 5.2b.

Since the snake makes $m$ moves before returning to $C$, we know that $s_\ell^- = s_{\ell-m}$. Thus, the length of $P_C$ is maximized if $s_1^+ = s_{\ell-m}$. In this case, $P_C$ consists of $(s_\ell, \ldots, s_{\ell-m})$ and the path from $s_1$ to $s_\ell$ through the section of $C$ that was unoccupied by $S$. The section $(s_\ell, \ldots, s_{\ell-m})$ has length $m$. Before the head leaves $C$, there are $|C| - \ell$ unoccupied vertices between $s_1$ and $s_\ell$. This forms a path of length $|C| - \ell + 1$. Combined, we obtain that $P_C$ has length at most $|C| - \ell + 1 + m$.

The head moves from $s_1$ to $s_1^+$ by visiting $m$ vertices in $\overline{C}$. It follows that there must also exist some $s_1 s_1^+$-path of length at most $m + 1$ with all internal vertices in $\overline{C}$, as depicted in Figure 5.2b. We will call this path $P_{\overline{C}}$, as depicted in Figure 5.2a. By combining $P_C$ and $P_{\overline{C}}$, we obtain a cycle of length at most $|C| - \ell + 2m + 2$. It follows that $g(G) \leq |C| - \ell + 2m + 2$. □

**Corollary 5.3.** *Let $G = (V, E)$ be a graph with $g(G) > 2k$. Let $\ell = |V| - k$ be the length of the snake, with $k \geq 2$. Suppose $G$ has a cycle $C$ that contains the snake. If the head of the snake leaves $C$, then it must visit all vertices in $\overline{C}$ before returning to $C$.*

*Proof.* Let $m$ be the number of vertices that the head visits before returning to $C$. By Lemma 5.2, we know $g(G) \leq |C| - \ell + 2m + 2$. Suppose the head does not visit all the vertices in $\overline{C}$, in other words, we have $m \leq |V| - |C| - 1$. This gives us the following upper bound for $g(G)$:

$$g(G) \leq |C| - \ell + 2m + 2 \leq 2|V| - |C| - \ell.$$

Since we have $\ell = |V| - k$, it follows that

$$g(G) \leq |V| - |C| + k.$$

Since $C$ contained the snake, we know that $|C| \geq |V| - k$, and thus we have $|V| - |C| \leq k$. This gives us $g(G) \leq 2k$, which contradicts that $g(G) > 2k$. We conclude that the head must visit all vertices in $\overline{C}$ before returning to $C$. □

Of course, we might wonder why the snake would want to return to $C$. The following observation shows us that if there are unoccupied vertices on $C$, then the snake has to return at some point.

**Observation 5.4.** *Let $\overline{S}$ be the current set of unoccupied vertices. To win, the snake will have to visit all vertices in $\overline{S}$ at some point in the future.*

*Proof.* To win, the snake has to obtain a position that occupies all vertices. If a vertex is currently unoccupied, then it can only become occupied if the snake visits it. □

Whenever the snake eats an apple, the tail remains in place. So if the snake eats an apple on one of the $m$ vertices it visits in $\overline{C}$, then the maximal length of the section it skips becomes one shorter. If the apple is placed on some vertex in $\overline{C}$ and the snake has to visit all vertices in $\overline{C}$ as in Corollary 5.3, then the snake has to eat the apple before it can return to $C$. We can use this to prove the following lemma.

**Lemma 5.5.** *Let $G = (V, E)$ be a graph with $g(G) > 2k$. Let $\ell = |V| - k$ be the length of the snake with $k \geq 2$. Suppose $G$ has a cycle $C$ that contains the snake with $|C| > \ell$. If the apple is on some vertex in $\overline{C}$, then the snake will lose.*

*Proof.* Suppose the apple is on some vertex in $\overline{C}$. Then at some point, the head of the snake will have to leave $C$ to eat the apple. We will first show that after the head leaves $C$, it can never return to $C$.

Suppose the head leaves $C$ and returns to $C$ at some later point. By Corollary 5.3, the snake has to visit all vertices in $\overline{C}$ before it can return. Let $m$ be the number of vertices the snake visits before returning to $C$, then $m = |V| - |C|$. Furthermore, one of these vertices must contain the apple.

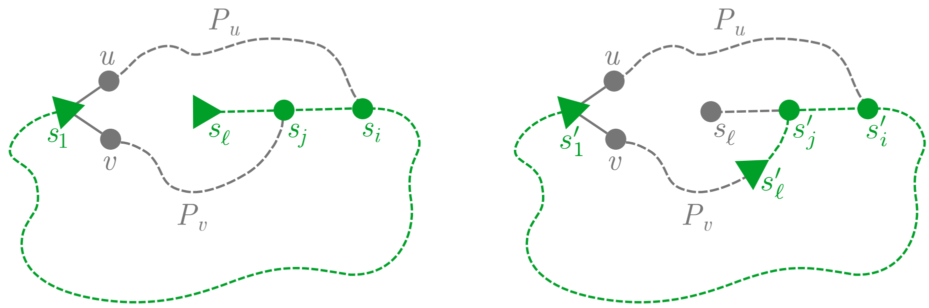
When the snake eats the apple, the tail will not move. So while the snake visits $m$ vertices in $\overline{C}$, the tail will only move $m - 1$ times. It follows that right before the head moves back to $C$, the tail will be on $s_{\ell-m+1}$. Let $s_1^+$ be the position of the head right after it re-enters $C$. We now use similar reasoning as for Lemma 5.2 to obtain that there is a path on $C$ from $s_1$ to $s_1^+$ of length at most $|C| - \ell + m + 1$.

Combined with the path of the head through $\overline{C}$, this gives us a cycle of length at most $|C| - \ell + 2m + 1$. Given that $m = |V| - |C|$, we obtain that

$$|C| - \ell + 2m + 1 = 2|V| - |C| - \ell + 1.$$

Since $|C| \geq \ell + 1$, we have

$$2|V| - |C| - \ell + 1 \leq 2(|V| - \ell),$$




(a) The snake can take $P_u$ or $P_v$ to $S$. (b) The snake in position $S'$.

Figure 5.3: There is no cycle that contains the snake and the head can move to either $u$ or $v$ next, without ensuring a loss.

and since $\ell \geq |V| - k$, it follows that $G$ must contain a cycle of length at most $2k$. This contradicts that $g(G) > 2k$, and thus we can conclude that the head cannot return to $C$.

Since $|C| \geq \ell + 1$, there must be some vertex $v$ on $C$ that is unoccupied right before the head leaves $C$. Since the snake can no longer return to $C$, the snake can never visit $v$ again, and $v$ will remain unoccupied. By Observation 5.4, the snake will lose. □

**Corollary 5.6.** *Let $G = (V, E)$ be a graph with $g(G) > 6$. Consider the moment the snake grows to length $|V| - 3$. If there is a cycle $C$ of length $|V| - 1$ or $|V| - 2$ that contains the snake, then the snake will lose.*

*Proof.* Since the snake is contained in $C$ and $|C| < |V|$, there is some unoccupied vertex in $\overline{C}$. The apple placer places the next apple on this vertex. We can now use Lemma 5.5 with $k = 3$ to obtain that the snake will lose. □

Corollary 5.6 excludes two out of the four possible scenarios for when the snake reaches length $|V| - 3$. The two that remain are: the snake is contained in a cycle of length $|V| - 3$, and there is no cycle that contains the snake. Hence, we still need to show both of these scenarios will result in a loss for the snake. For the latter case, the following lemma will be useful.

**Lemma 5.7.** *Let $G = (V, E)$ be a graph with $g(G) > 2k$ and let $\ell = |V| - k$ be the length of the snake with $k \geq 2$. Suppose there is no cycle in $G$ that contains the snake. Then, there is at most one vertex to which the snake can move next without ensuring a loss.*

*Proof.* The idea behind the proof is as follows. If there are two different vertices the snake can move to, then this will give us two different paths through $\overline{S}$. We will see that these paths must both lead to $S$ and cannot meet before they reach $S$. With one of these paths, we then construct a cycle that contains a big portion of $S$. Using this cycle and the remaining path, we will show that the snake can move in a way that contradicts Corollary 5.3.

Let $S$ be the current position of the snake with $s_1$ the head position, and $s_\ell$ the tail position. Let $u$ and $v$ be two unoccupied vertices that are both adjacent to the head. Suppose the snake can move to either $u$ or $v$ next, without ensuring a loss. First, suppose the snake moves to $u$. Then it cannot move to $v$ before the head moves to $S$ again, otherwise we would have a cycle of length at most $k+1$. By Observation 5.4 the snake will eventually have to visit $v$ to win. It follows that the snake has to move to $S$ at some point. With the same argument, this is also the case if the snake moves to $v$ first.

Let $P_u$ and $P_v$ be the paths the head can take from $u$ and $v$ to $S$ respectively. The two paths are depicted in Figure 5.3a. We know $P_u$ and $P_v$ must be internally disjoint, otherwise, we would have a cycle of length at most $k+1$. Let $s_i$ be the endpoint of $P_u$ and $s_j$ of $P_v$. Without loss of generality, we may assume that $i \leq j$. Since there is no cycle that contains $S$, we must have $i, j < \ell$. This means $s_\ell$ lies on neither of the two paths.

Suppose the snake takes the path $P_v$. When the head moves to $s_j$, the cycle $C = (s_{j-1}, \dots s_1, P_u)$ will contain the snake. The snake can now keep moving along $C$ until the head is on $s_1$ again. Let $S'$ be the position of the snake at this moment, which is depicted in Figure 5.3b. Note that $C$ still contains $S'$. This new position is similar to the original position $S$, but the last section of the snake, from $s'_{j+1}$ to $s'_l$, now lies on $P_v$. Since we assumed the snake could also take $P_u$ from $u$ to $s_i$, it follows that from the position $S'$, the snake can take the path $P_u$ to leave and re-enter $C$. However, we know $s_\ell$ neither lies on this path, nor on the cycle $C$. By Corollary 5.3, this is not possible. It follows that from the original position $S$, either moving to $u$ or moving to $v$ must result in a loss. □

By Lemma 5.7, we obtain that if $G$ has a girth of at least 7, the snake has length $|V|-3$, and there is no cycle containing the snake, then we can predict its movement. We can use this to our advantage when describing a strategy for the apple placer. The following lemma will be useful for this purpose.

**Lemma 5.8.** *Let $G = (V, E)$ be a non-Hamiltonian graph with $g(G) > 4$. When the snake grows to length $|V| - 2$, if the two unoccupied vertices are not adjacent, then the snake will lose.*

*Proof.* Since $G$ is non-Hamiltonian, there is either a cycle of length $|V| - 2$, a cycle of length $|V| - 1$, or no cycle that contains the snake. By Lemma 5.5, the snake will lose if there is a cycle of length $|V| - 1$ that contains the snake.

First, suppose that there is a cycle of length $|V| - 2$ that contains the snake. In other words, the head of the snake is adjacent to the tail, and the snake itself forms a cycle. Let $C$ be this cycle. Both unoccupied vertices are in $\overline{C}$, and the apple has to be placed on one of these two vertices. The snake cannot eat the apple by only moving along $C$. Hence, at some point, it must move to one of the unoccupied vertices. When it does so, by Corollary 5.3, the snake has to visit the other unoccupied vertex next. However, this is not possible, since the unoccupied vertices are not adjacent. It follows that the snake will lose.

Now suppose there is no cycle that contains the snake. By Lemma 5.7 there is only one unoccupied vertex the head can move to without losing. The apple is placed on this vertex, forcing the snake to eat the apple on its next move. From there, the head cannot move to its tail, since there was no cycle that contained the snake. It can also not move to the remaining unoccupied vertex since they are not adjacent. It follows that the snake will lose. □

We now return to our scenario where $G$ is non-Hamiltonian with $g(G) > 6$, the snake grows to length $|V| - 3$, and there is no cycle containing the snake. By Lemma 5.8, the apple placer only needs to ensure that when the snake eats the next apple, the remaining unoccupied vertices are not adjacent. This allows us to prove the following.

**Lemma 5.9.** *Let $G = (V, E)$ be a non-Hamiltonian graph with $g(G) > 6$. Consider the moment the snake grows to length $\ell = |V| - 3$ and suppose there is no cycle in $G$ that contains the snake. Then, the snake will lose.*

*Proof.* Let $S$ be the current snake position with $s_1$ and $s_\ell$ being the head and tail positions respectively. We will distinguish between two cases. In the first case, the snake is able to visit all three unoccupied vertices without moving to $S$ in between. In the second case, this is not possible.

**Case 1: the snake can visit all vertices in $\overline{S}$ without moving to $S$ in between.**

First, we note that by Lemma 5.7, the snake can visit the vertices in $\overline{S}$ in only one order. Let $u$ be the vertex the snake has to move to first, $v$ the second, and $w$ the third. The apple is placed on $v$. First, the snake has to move to $u$, after which $s_\ell$ becomes unoccupied. Next, the snake has to eat the apple on $v$. After eating the apple, $w$ and $s_\ell$ are the two remaining unoccupied vertices. But $w$ cannot be adjacent to $s_\ell$ since this would give us the cycle $(S, u, v, w)$, contradicting that $G$ is non-Hamiltonian. Hence, the two remaining unoccupied vertices are not adjacent and by Lemma 5.8, the snake will lose.

**Case 2: the snake cannot visit all vertices in $\overline{S}$ without moving to $S$ in between.**

By Lemma 5.7, there is at most one vertex the head can move to without guaranteeing a loss. Let this be vertex $u$. The apple is placed on $u$, meaning the snake has to eat the apple on its next move. After eating the apple on $u$, we know the snake cannot move to $s_\ell$, since then the cycle $(S, u)$ would then have contained $S$. Thus, one of the remaining unoccupied vertices has to be adjacent to $u$, otherwise the snake will lose. Let $v$ be an unoccupied vertex that is adjacent to $u$ and let $w$ be the other unoccupied vertex. Since we assumed the snake cannot visit all vertices in $\overline{S}$ without moving to $S$ in between, the snake should not be able to move from $u$ to $v$ and then to $w$. It follows that the two remaining unoccupied vertices $v$ and $w$ cannot be adjacent and by 5.8, the snake will lose. □

From our four possible scenarios, we have now shown three will result in a loss for the snake. It remains to show that if there is a cycle of length $|V| - 3$ that contains the snake, then the apple placer has a winning strategy. Note that in this scenario, the snake itself forms is cycle and the head is adjacent to the tail.

**Lemma 5.10.** *Let $G = (V, E)$ be a non-Hamiltonian graph with $g(G) > 6$. Consider the moment the snake grows to length $\ell = |V| - 3$ and suppose there is a cycle $C$ of length $|V| - 3$ that contains the snake. Then, the snake will lose.*

*Proof.* Since $C$ has the same length as the snake, the vertices of $C$ are exactly those that are occupied by the snake. This also means all three unoccupied vertices are in $\overline{C}$. Note that the three unoccupied vertices cannot form a cycle by themselves, since $G$ does not contain any cycles of length three. Hence, there is at most one unoccupied vertex that is adjacent to both of the other unoccupied vertices. If such a vertex exists, the apple is placed on it. We will refer to the vertex with the apple as vertex $a$.

The snake cannot eat the apple by only moving along $C$. Hence, at some point, it must move to an unoccupied vertex. By Corollary 5.3, we know that once the snake leaves $C$, it cannot return to $C$ without visiting all unoccupied vertices first. Furthermore, by Observation 5.4, the snake cannot win without visiting all three unoccupied vertices. But since $a$ is the only vertex that can be adjacent to both other unoccupied vertices, it has to be the second unoccupied vertex the snake visits.

The scenario when the snake leaves $C$ is depicted in Figure 5.4. Let $u$ be the unoccupied vertex the snake moves to before moving to $a$, and let $v$ be the remaining unoccupied vertex. Let $s_\ell$ be the tail position right before the head moves to $u$. When the head moves to $u$, $s_\ell$ becomes unoccupied. When the snake eats the apple on $a$, $s_\ell$ remains unoccupied, and $v$ and $s_\ell$ are the two remaining unoccupied vertices. By Lemma 5.3, the head has to move to $v$ next, and thus $v$ must be adjacent to $a$. But then, $v$ cannot be adjacent to $s_\ell$, since this would create the cycle $(S, u, a, v)$ contradicting that $G$ is non-Hamiltonian. Hence, $v$ and $s_\ell$, the two remaining unoccupied vertices, are not adjacent, and by Lemma 5.8 the snake will lose. □

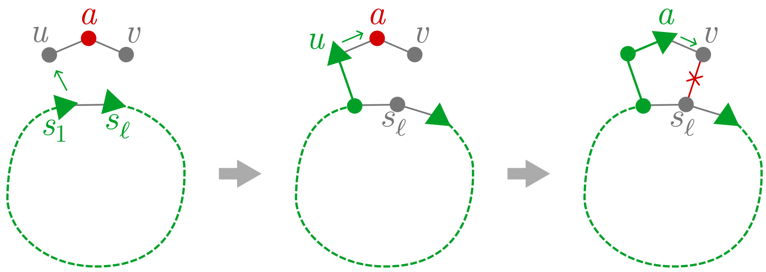



Figure 5.4: From a cycle of length $|V| - 3$, the snake moves to $u$ before eating the apple on $a$. If the snake can move to $v$, the remaining two unoccupied vertices cannot be adjacent.

We can now prove the following upper bound on the girth of non-Hamiltonian snake-winnable graphs.

**Theorem 5.11.** *Let $G = (V, E)$ be a non-Hamiltonian graph with $g(G) > 6$. Then, $G$ is not snake-winnable.*

*Proof.* Consider the moment the snake grows to length $|V| - 3$. Since $G$ is non-Hamiltonian, there cannot be any cycle of length $|V|$. Hence, the snake is either contained in a cycle of length $|V| - 1$, a cycle of length $|V| - 2$, a cycle of length $|V| - 3$, or in no cycle.

If the snake is contained in a cycle of length $|V| - 1$ or $|V| - 2$, then the snake will lose by Corollary 5.6. If the snake is contained in a cycle of length $|V| - 3$, then it will lose by Lemma 5.10. Finally, if the snake is not contained in any cycle, then it will lose by Lemma 5.9. It follows that once the snake reaches length $|V| - 3$, the apple placer can guarantee the snake will lose, and thus $G$ is not snake-winnable. □

## 5.2 The girth of snake-winnable grid graphs and partial grid graphs

From Theorem 5.11, the question arises whether there are any non-Hamiltonian snake-winnable graphs with a girth of 6. We will show that such a graph does exist. But first, we examine the implications Theorem 5.11 has for grid graphs.

**Observation 5.12.** *A grid graph cannot have a girth of 6.*

*Proof.* In a grid graph, a cycle of length 6 cannot be cordless, as depicted in Figure 5.5. Hence, if a grid graph has a cycle of length 6, then it must also have a cycle of length 4. □

From Observation 5.12, we obtain the following.

**Lemma 5.13.** *All non-Hamiltonian snake-winnable grid graphs have a girth of 4.*

*Proof.* By Theorem 5.11, a non-Hamiltonian snake-winnable grid graph cannot have a girth of 7 or higher. By Observation 5.12, it can also not have a girth of 6. Since any grid graph is bipartite, it can only contain cycles of even length and therefore the girth cannot be odd. It follows that the only girth a non-Hamiltonian snake-winnable grid graph can have is 4. □

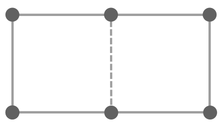


Figure 5.5: In a grid graph, a cycle of length 6 cannot be cordless.

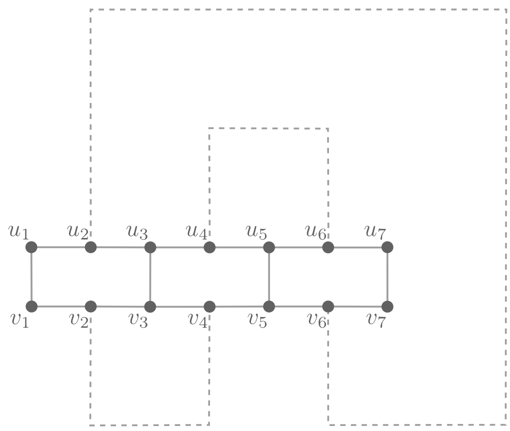



Figure 5.6: A non-Hamiltonian partial grid graph with a girth of 6 that is snake-winnable.

If we want to find an example that shows the bound from Theorem 5.11 is tight, then this cannot be a grid graph. Hence, we turn to the more general class of partial grid graphs.

**Definition 5.14.** *A graph $G = (V, E)$ is a* ***partial grid graph*** *if and only if it is a subgraph of some rectangular grid graph.*

Since partial grid graphs do not have to be a vertex-induced subgraph of a rectangular grid graph, they can have cycles of length 6 that are cordless. The graph in Figure 5.6 is non-Hamiltonian, snake-winnable, and has a girth of 6.

**Lemma 5.15.** *There are non-Hamiltonian partial grid graphs with a girth of 6 that are snake-winnable.*

*Proof.* Note that the graph in Figure 5.6 is a spanning subgraph of the graph in Figure 3.7. Since this graph is non-Hamiltonian, the graph in Figure 5.6 must be non-Hamiltonian as well. Furthermore, it also contains the cycles $C_1$ and $C_2$, as depicted in Figure 5.7. Hence, we can use the exact same strategy for the snake as for Lemma 3.16. □

Lemma 5.15 shows us that the bound from Theorem 5.11 is tight, even on the class of partial grid graphs. Since partial grid graphs are bipartite and planar, the bound is tight on those classes as well.

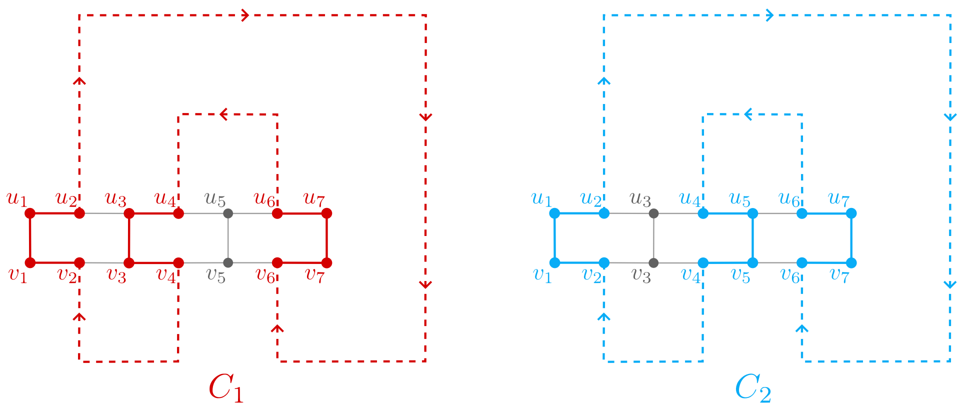



Figure 5.7: The cycles $C_1$ and $C_2$.

# 6. The structure of snake-winnable graphs

In this section, we will provide a necessary structure for all non-Hamiltonian snake-winnable graphs. For this, we first examine a different variant of the game where the head is never allowed to move to the tail vertex.

## 6.1 The Head Moves First variant

In Section 2 we defined the rules for the movement of the snake as follows. If the snake is in the position $S = (s_1, \ldots, s_\ell)$, then the head has to move to some vertex in $N(s_1) \cap \big(\overline{S} \cup \{s_\ell\}\big)$. We can distinguish three different types of moves, which are depicted in Figure 6.1.

In the first type of move, the head moves to the apple and the tail remains in place. This means one vertex is added to the snake and, consequentially, one vertex is removed from the unoccupied set. We will refer to this as a *type* $\alpha$ move.

In the second type of move, the snake moves to an unoccupied without the apple. In this case, the new head position is added to the snake, and the former tail is removed. This also means one vertex is removed from the unoccupied set and a different vertex is added. We will refer to this as a *type* $\beta$ move.

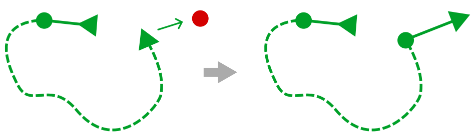
In the third move type, the head moves to the tail vertex and the unoccupied set remains unchanged. We will refer to this as a *type* $\gamma$ move.

**Definition 6.1.** *Let $S = (s_1, \ldots, s_\ell)$ be the position of the snake. In the* ***Head Moves First*** *variant, abbreviated as* ***HMF****, the head has to move to some vertex in $N(s_1)$. In other words, in the HMF variant, the snake can only make type $\alpha$ and type $\beta$ moves.*

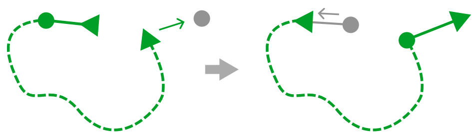
For a graph, if the snake has a winning strategy for the HMF variant, then we will call it *HMFsnake-winnable*

**Observation 6.2.** *If a graph is HMFsnake-winnable, then it is also snake-winnable.*

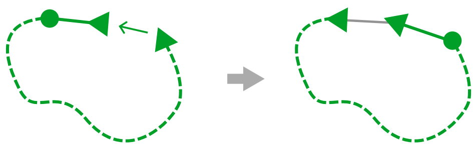
*Proof.* If a graph is HMFsnake-winnable, then the snake has a winning strategy that only consists of type $\alpha$ and type $\beta$ moves. Since both of these move types are also allowed in the (normal) game of Snake, we can use the same strategy as a winning strategy for Snake. □



(a) A type $\alpha$ move.



(b) A type $\beta$ move.



(c) A type $\gamma$ move.

Figure 6.1: The three different types of moves.

The reverse, however, does not hold. In fact, we will show that the HMFsnake-winnable graphs are exactly those that are Hamiltonian. By Observation 2.3, we saw any Hamiltonian graph is snake-winnable. The same holds for the HMF variant.

**Observation 6.3.** *If $G = (V, E)$ is Hamiltonian, then $G$ is HMFsnake-winnable.*

*Proof.* Since $G$ is Hamiltonian, there is some simple cycle $C$ in $G$ that contains all vertices in $V$. Just like for the normal variant, the snake can keep moving along this cycle. Since $C$ has length $|V|$, until the snake reaches length $|V|$, there will always be some unoccupied vertex on $C$ between the head and the tail. This means the snake never has to make a type $\gamma$ move to move along the cycle, and thus $G$ is HMFsnake-winnable. □

If a vertex is not on the snake and does not contain the apple, then we will call it an *empty vertex*. To show that non-Hamiltonian graphs are not HMFsnake-winnable, we will use the following two observations.

**Observation 6.4.** *After the snake makes a type $\beta$ move, there is an empty vertex that is adjacent to the tail.*

*Proof.* Let $S^t = (s_1^t, \dots, s_\ell^t)$ be the position of the snake. If the snake makes a type $\beta$ move, then the head moves to some empty vertex $v \in \overline{S^t}$. The next snake position is $s^{t+1} = (v, s_1^t, \dots, s_{\ell-1}^t)$. Thus, $s_\ell^t$ becomes unoccupied and is adjacent to the new tail $s_{\ell-1}^t$. By Observation 2.1, $s_\ell^t$ cannot contain the apple, and thus must be empty. □

**Observation 6.5.** *Suppose $v$ is empty and the snake makes a type $\alpha$ move to vertex $a$. Then, $v$ remains unoccupied after this move.*

*Proof.* First note that $a \neq v$, since $v$ is empty. Let $\overline{S^t}$ be the unoccupied set at time $t$, with $v \in \overline{S^t}$. If the snake makes a type $\alpha$ move, then $\overline{S^{t+1}} = \overline{S^t} \backslash \{a\}$. Hence, we have $v \in \overline{S^{t+1}}$. □

Using Observations 6.4 and 6.5, we can prove the following lemma, which plays an important role in the apple placer's strategy.

**Lemma 6.6.** *Suppose the snake plays by the rules of the HMF variant and makes a type $\beta$ move. Then the apple placer can ensure that when the snake reaches length $|V| - 1$, the remaining unoccupied vertex is adjacent to the tail.*

*Proof.* First note that the snake can only make a type $\beta$ move if it has length at most $|V|-2$, since at length $|V| - 1$ the only remaining unoccupied vertex contains the apple.

After the snake makes the type $\beta$ move, by Observation 6.4, there is some empty vertex that is adjacent to the tail. Since the snake can only make type $\beta$ moves until it eats the apple, this remains the case until the snake makes a type $\alpha$ move. So right before the snake makes the next type $\alpha$ move, there is some vertex $v$ that is empty and adjacent to the tail. By Observation 6.5, directly after the type $\alpha$ move, $v$ remains unoccupied. Since the tail does not move during a type $\alpha$ move, $v$ also remains adjacent to the tail. If the snake has not yet reached length $|V| - 1$, then the apple placer can place the apple on some vertex $a \neq v$. This means $v$ remains empty and adjacent to the tail. If the snake immediately makes another type $\alpha$ move, then by Observation 6.5, $v$ remains unoccupied and adjacent to the tail. If the snake makes a type $\beta$ move next, then by Observation 6.4, after this move there is an empty vertex that is adjacent to the tail. It follows that the apple placer can repeat this strategy until the snake reaches length $|V| - 1$. □

We will use Lemma 6.6 to formulate a winning apple placer strategy for the HMF variant on non-Hamiltonian graphs.

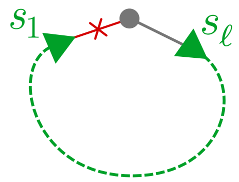



Figure 6.2: If $G$ is non-Hamiltonian and the only unoccupied vertex is adjacent to the tail, then the head cannot be adjacent to the unoccupied vertex.

**Theorem 6.7.** *A graph is HMFsnake-winnable if and only if it is Hamiltonian.*

*Proof.* By Observation 6.3, we know that if a graph is Hamiltonian, then it is HMFsnake-winnable. It remains to show that if a graph is non-Hamiltonian, then it is not HMFsnake-winnable.

Let $G = (V, E)$ be a non-Hamiltonian graph. Since $G$ is non-Hamiltonian, there must be two vertices $u, v \in V$ that are not adjacent to each other. The first apple is placed on $u$ and the second on $v$. Since the snake cannot immediately move from $u$ to $v$ it must make a type $\beta$ move before eating the apple on $v$. By Lemma 6.6, it follows that the apple placer can ensure that when the snake reaches length $|V| - 1$, the remaining unoccupied vertex is adjacent to the tail, as depicted in Figure 6.2.

Since this is the only remaining unoccupied vertex and the snake is following the rules of the HMF variant, it must move to this vertex. However, by doing so it would create a Hamiltonian cycle. Since we assumed $G$ was non-Hamiltonian, it follows that after it reaches length $|V| - 1$, there is no move the snake can make, and thus the snake will lose. □

## 6.2 A necessary structure for snake-winnable graphs

From Theorem 6.7, we see that type $\gamma$ moves are essential for snake strategies on non-Hamiltonian graphs. However, the snake might not always be able to make such a move.

**Definition 6.8.** *The **circumference** of a graph $G$ is the length of the longest simple cycle in $G$ and is denoted as is denoted circ$(G)$.*

**Observation 6.9.** *Let $\ell$ be the current length of the snake on $G = (V, E)$. If $\ell >$ circ$(G)$, then the snake can no longer make type $\gamma$ moves.*

*Proof.* To make a type $\gamma$ move, the head of the snake has to be adjacent to the tail. This means the snake will form a cycle right before and after a type $\gamma$ move, which is only possible if $G$ contains a cycle of length $\ell$. □

Note that if $G = (V, E)$ is non-Hamiltonian, then $\mathrm{circ}(G) < |V|$. Hence, for non-Hamiltonian graphs, there will be some point at which the snake length exceeds $\mathrm{circ}(G)$ and can no longer make type $\gamma$ moves. In essence, the snake will be playing by the HMF variant rules from this point on. So if the apple placer can force a type $\beta$ move after this point, then by Lemma 6.6, the snake will lose. We will use this to describe a necessary structure for snake-winnable graphs.

**Definition 6.10.** *Let $S^t = (s_1^t, \ldots, s_\ell^t)$ be the position of the snake on $G$ at time $t$. The **head graph** at time $t$, denoted $H^t$, is the subgraph of $G$ induced by $\overline{S^t} \cup \{s_1^t\}$.*

An example of the head graph can be seen in Figure 6.3.

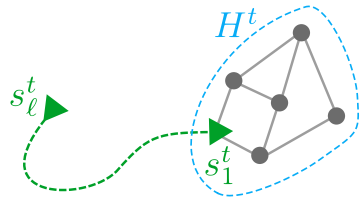



Figure 6.3: The head graph $H^t$ is the subgraph induced by $\overline{S^t} \cup s_1^t$.

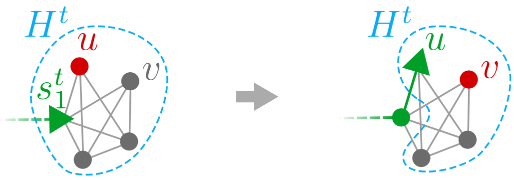



Figure 6.4: The vertices $u$ and $v$ are not adjacent to each other. If the snake immediately eats the apple on $u$, then it has to make a type $\beta$ move to reach $v$.

**Lemma 6.11.** *Let $G = (V, E)$ be a non-Hamiltonian graph and let $t$ be the moment the snake reaches length circ$(G) + 1$. If $H^t$ is incomplete, then the apple placer can force the snake to make a type $\beta$ move within the next two moves.*

*Proof.* Since the length of the snake is greater than circ$(G)$, the snake can only make type $\alpha$ and type $\beta$ moves. Let $s_1^t$ be the head position of the snake at time $t$. We distinguish between two different cases: there is some vertex $v$ in $H^t$ that is not adjacent to $s_1^t$, or all vertices in $H^t$ are adjacent to $s_1^t$.

First, suppose there is some vertex $v \in \overline{S^t}$ that is not adjacent to $s_1^t$. The apple placer places the next apple on $v$. Since the head cannot directly move from $s_1$ to $v$, it has to make a type $\beta$ move next.

Next, suppose all vertices in $\overline{S^t}$ are adjacent to $s_1^t$, as depicted in Figure 6.4. Since $H^t$ is incomplete, there must be two vertices $u, v$ in $\overline{S^t}$ that are not adjacent to each other. The apple placer places the first apple on $u$. If the snake does not make a type $\beta$ move next and immediately moves to $u$, then $v$ remains unoccupied and the apple placer can place the next apple on $v$. Since $v$ is not adjacent to $u$, the snake will have to make a type $\beta$ move next. □

**Corollary 6.12.** *Let $G = (V, E)$ be a non-Hamiltonian snake-winnable graph and let $t'$ be the moment the snake reaches length circ$(G)+1$. Then for all $t \geq t'$, $H^t$ must be complete.*

*Proof.* Suppose $H^t$ is incomplete. Since the length of the snake is greater than circ$(G)$, the snake can only make type $\alpha$ and type $\beta$ moves after time $t$. By Lemma 6.11, the snake can be forced to make a type $\beta$ move within the next two moves. It follows by Lemma 6.6 that the snake will lose. □

Corollary 6.12 shows us that for non-Hamiltonian snake-winnable graphs, the head graph must remain complete during the last part of the game.

**Definition 6.13.** *Let $t$ be the first time the snake eats an apple and the head graph is complete. We refer to the phase of the game starting from $t$ onward as the* ***complete phase.***

Note that the complete phase can only start when the snake eats an apple. Thus, if the complete phase starts at time $t$, then the snake made a type $\alpha$ move from $t-1$ to $t$. It turns out that once the snake reaches the complete phase, it is guaranteed to win.

**Observation 6.14.** *Suppose at some time $t$, $H^t$ is complete. Then, the snake will win.*

*Proof.* Let $s_1^t$ be the head of the snake. Since $H^t$ is complete, the apple $a$ must be adjacent to $s_1^t$. The snake can immediately move to $a$, and for the next head graph, we get $H^{t+1} = H^t - s_1^t$. Hence, the head graph $H^{t+1}$ remains complete. The snake can keep repeating this strategy until it reaches length $|V|$. □

Since reaching the complete phase results in a guaranteed win for the snake, the apple placer will always try to prevent this. This gives us the following.

**Lemma 6.15.** *Let $S^t = (s_1^t, \ldots, s_\ell)$ be the position of the snake at time $t$, and suppose the snake made a type $\alpha$ move from time $t-1$ to $t$. Furthermore, suppose $H^t$ is incomplete. Then the snake cannot enter the complete phase at time $t+1$.*

*Proof.* Suppose the snake can enter the complete phase at time $t+1$. Then the snake has to eat an apple at time $t+1$, and thus the apple must be placed in $N(s_1^t)$. We then get that $H^{t+1} = H^t - s_1^t$ is complete. Since $H^t$ was incomplete, there was some vertex in $\overline{S^t}$ that is not adjacent to $s_1^t$. Hence, the apple placer could have placed the apple on a vertex that is not in $N(s_1^t)$. By Observation 6.14, the snake will win if it reaches the complete phase. Thus, the apple placer always wants to prevent the snake from entering the complete phase. We conclude that the apple placer would never have placed the apple in $N(s_1^t)$, and thus the snake cannot enter the complete phase at time $t+1$. □

Lemma 6.15 shows us that the snake cannot enter the complete phase right after making a type $\alpha$ move. We will now show that on non-Hamiltonian graphs, the snake can also not enter the complete phase right after a type $\beta$ move.

**Lemma 6.16.** *Let $G$ be non-Hamiltonian. If the snake makes a type $\beta$ move followed by a type $\alpha$ move, then after the type $\alpha$ move, the head graph cannot be complete.*

*Proof.* By Observation 6.4 and Observation 6.5, after the type $\alpha$ move, there is an empty vertex that is adjacent to the tail. Let this be vertex $v$. Let $S = (s_1, \ldots, s_\ell)$ be the position of the snake, and $H$ the head graph after the type $\alpha$ move. Note that $H$ contains both $v$ and $s_1$. If $H$ is complete, then it contains a path $P$ from $s_1$ to $v$ that contains all vertices in $H$. But since $v$ is adjacent to $s_\ell$, we can form a Hamiltonian cycle in $G$ by combining $P$ and $S$. This contradicts that $G$ is non-Hamiltonian, and thus $H$ cannot be complete. □

**Corollary 6.17.** *Let $G$ be a non-Hamiltonian graph. If the snake enters the complete phase at time $t$, then from $t-2$ to $t-1$, the snake must have made a type $\gamma$ move.*

*Proof.* By Lemma 6.15, the snake could not have made a type $\alpha$ move from $t-2$ to $t-1$. By Lemma 6.16, the snake could also not have made a type $\beta$ move from $t-2$ to $t-1$. Hence, the snake must have made a type $\gamma$ move from $t-2$ to $t-1$. □

After a type $\gamma$ move, the head of the snake will always be adjacent to the tail. Thus, Corollary 6.17 shows us that the snake always forms a cycle right before it enters the complete phase. If the snake enters the complete phase at time $t$, then we will refer to $V(H^t)$ as the *final clique* and the cycle formed by $S^{t-1}$ as the *final cycle*.

Corollary 6.17 gives us the following necessary structure for snake-winnable graphs.

**Theorem 6.18.** *Let $G = (V, E)$ be a snake-winnable graph. Then $G$ must contain a clique $Q$ of size at least $|V| - circ(G)$. Furthermore, there must be a cycle that $C$ in $G$ such that $\overline{C} = Q$.*

*Proof.* First note that if $G$ is Hamiltonian, then the statement is trivially true since $\text{circ}(G) = |V|$, and thus we can have $Q = \emptyset$.

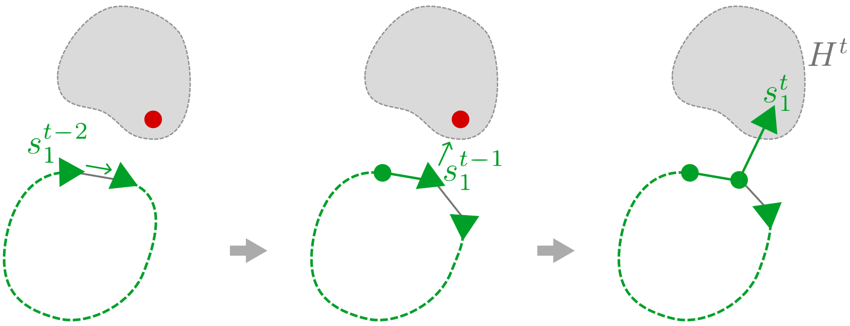



Figure 6.5: The snake makes a type $\gamma$ move from $t-1$ to $t-2$, and enters the complete phase at time $t$. We get that $S^{t-1}$ forms a cycle a $V(H^t)$ is a clique.

Suppose $G$ is not Hamiltonian. By Corollary 6.12, there must be some time $t$ at which the snake enters the complete phase. Let $S^t = (s_1^t, \ldots, s_\ell^t)$ be the position of the snake at time $t$. We have the clique $Q = V(H^t)$ with $|Q| = |V| - \ell + 1$. By Corollary 6.17, the snake makes a type gamma move from $t-2$ to $t-1$, as depicted in Figure 6.5. Let $C$ be the cycle formed by $S^{t-1}$. Note that $C$ has length $\ell - 1$. Furthermore, $V(H^t) = \overline{C}$ and thus

$$|Q| = |V| - |C| = |V| - \ell + 1 \geq |V| - \mathrm{circ}(G).$$

□

For even-sized bipartite graphs, we saw that having a $\Theta(|V|-5, 3, 3)$ spanning subgraph was not sufficient for it to be snake-winnable. Note that having a $\Theta(|V|-5, 3, 3)$ spanning subgraph satisfies the structural requirement form Theorem 6.18 with a cycle of length $|V|-2$ and a clique of size 2. Hence, we can also conclude that while this structural requirement is necessary, it is not sufficient for a graph to be snake-winnable.

# 7. Snake on graphs with low vertex-connectivity

In Section 2, we specified that the game of Snake is always played on a connected graph. On a disconnected graph, the apple could always be placed in a different component than the head, making it trivially unwinnable for the snake. On connected graphs, there will always be a path from the head to the apple, but it might be blocked by the body of the snake. If the graph has low vertex-connectivity, then the number of paths to the apple may be limited. Intuitively, this makes it more likely for the snake to "block" itself, making it easier for the apple placer to win.

**Definition 7.1.** *Let $G = (V, E)$ be a connected simple graph. Let $S \subset V$ be a set of vertices such that $G - S$ is disconnected. Then $S$ is called a **vertex cut**. If there is a single vertex $v$ such that $G - v$ is disconnected, then we call $v$ a **cut vertex**.*

**Definition 7.2.** *Let $G = (V, E)$ be a connected simple graph. The **vertex-connectivity** of $G$, denoted $\kappa(G)$, is the size of the smallest vertex cut in $G$. For $K_n$, the complete graph on $n$ vertices, we define $\kappa(K_n) = n - 1$.*

For a graph G, we denote the number of connected components of $G$ by $\omega(G)$. Note that if $G$ is connected, then $\omega(G) = 1$. We can make the following two observations.

**Observation 7.3.** *If $G$ has a vertex cut $S$ and $\omega(G - S) > |S|$, then $G$ is non-Hamiltonian.*

*Proof.* Suppose $G$ does have a Hamiltonian cycle $C$. Let $v$ be some vertex that is not in $S$. If we start at $v$ and follow the cycle, then we will visit every component and end up back in the component with $v$. It follows that we will have to move through $S$ at least $\omega(G - S)$ times. But since $\omega(G - S) > |S|$, this is not possible and we conclude that $G$ cannot be Hamiltonian. □

**Observation 7.4.** *If $G$ has a vertex cut $S$ and $\omega(G - S) > |S| + 1$, then $G$ is not snake-winnable.*

*Proof.* By Observation 2.3, a graph needs to have a Hamiltonian path to be snake-winnable. Any Hamiltonian path passes through $S$ at least $\omega(G - S) - 1$ times. □

## 7.1 Snake-winnable graphs with vertex-connectivity 1

A graph with vertex-connectivity 1 is a connected graph with a cut vertex. Graphs with vertex-connectivity 1 can never be Hamiltonian, since any cycle has vertex-connectivity 2. Consider some graph with a cut vertex. If the snake wants to move between different components, it will have to move through the cut vertex. But if the snake is long, then the cut vertex will remain occupied for a while. Hence, the snake cannot return to another component for some time. In other words, as the snake grows longer, it becomes increasingly hard for the snake to move between different components. We will see that in many cases, the apple placer can use this to obtain a winning strategy. But first, we show that there do exist snake-winnable graphs with vertex-connectivity 1.

**Lemma 7.5.** *Let $G = (V, E)$ be a graph with $\kappa(G) = 1$ and let $v$ be a cut vertex of $G$. Suppose that $G_1$ and $G_2$ are the only two connected components of $G - v$ with $|V(G_1)| = |V(G_2)| = m$ and $m \geq 2$. Furthermore, let $G_1 + v$ and $G_2 + v$ both be complete. Then $G$ is snake-winnable.*

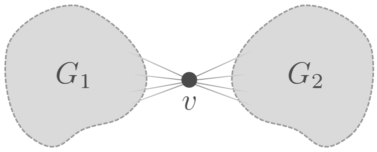



Figure 7.1: A snake-winnable graph with cut vertex $v$. The subgraphs $G_1 + v$ and $G_2 + v$ are both complete.

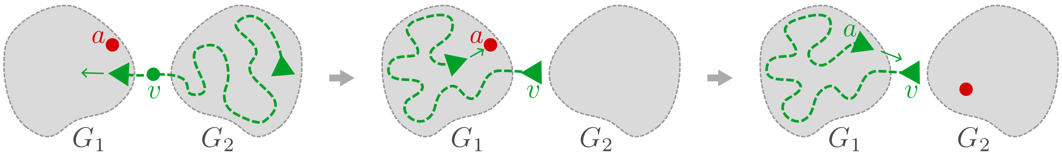



Figure 7.2: At length $m$, the snake moves into $G_1$ until its tail is on $v$. It then eats the apple on $a$, after which it occupies every vertex in $G_1$ and can move to $v$.

*Proof.* The graph $G$ is depicted in Figure 7.1. While the snake is shorter than $m$, it can always move between $G_1$ and $G_2$ by moving into the current component until $v$ becomes unoccupied, and then moving to the other component through $v$. We consider the moment the snake grows to length $m$.

First, suppose the next apple is placed on $v$. This means $v$ is currently unoccupied and thus the snake must be entirely in $G_1$ or entirely in $G_2$. By symmetry, we may assume the snake is entirely in $G_1$. Since the snake has length $m$, it occupies all the vertices in $G_1$. Furthermore, since $G_1 + v$ is complete the head must be adjacent to $v$. The snake can eat the apple on $v$ on the next move, after which the head graph will be $G_2 + v$, which is complete. By Observation 6.14 the snake will win.

Next, suppose after the snake grows to length $m$, the next apple is not placed on $v$. By symmetry, we may assume the apple is placed on some vertex $a$ in $G_1$, as depicted in Figure 7.2. If the head is in $G_2$, then the snake keeps moving into $G_2$ until $v$ is unoccupied. It then first moves its head to $v$, and then into $G_1$. With its head in $G_1$ the snake can now repeatedly move to vertices in $G_1$ that are not $a$, until its tail is on $v$. Since the snake has length $m$, it now occupies all vertices in $G_1$, except for $a$. The snake then eats the apple on $a$ and since $G_1 + v$ is complete, its head on $a$ will be adjacent to its tail on $v$. The head moves to $v$, after which the head graph is $G_2 + v$, which is complete. By Observation 6.14, the snake will win. □

It turns out the graph from Lemma 7.5 is the only type of snake-winnable graph with vertex-connectivity 1. To show this, we first prove the following two lemmas that show the limitations of the snake on graphs with vertex-connectivity 1.

**Lemma 7.6.** *Let $G = (V, E)$ be a graph with $\kappa(G) = 1$ and let $v$ be a cut vertex of $G$. Let $G_1$ and $G_2$ be two different connected components of $G - v$, with $|V(G_1)| = m$. Suppose the head of the snake is in $G_1$ and the snake has a length of at least $m + 2$. If there is an unoccupied vertex in $G_2$, then the snake will lose.*

*Proof.* By Observation 5.4, the snake will have to move to the unoccupied vertex in $G_2$ at some point in the future. Hence, at some point, the head will have to move from $G_1$ to $G_2$. To do so, the head will have to move through $v$. Thus, the snake must first keep moving into $G_1$ until it is either entirely in $G_1$, or only its tail outside of $G_1$, namely on $v$. But this is impossible since $G_1$ only contains $m$ vertices and the snake has a length of at least $m + 2$. □

**Lemma 7.7.** *Let $G = (V, E)$ be a graph with $\kappa(G) = 1$ and let $v$ be a cut vertex of $G$. Let $G_1$ and $G_2$ be two different connected components of $G - v$, with $|V(G_1)| = m$. Suppose the head of the snake is in $G_1$, the snake has a length of at least $m + 1$, and the apple is on some vertex $a$ in $G_1$. If there is some unoccupied vertex in $G_2$, then the snake will lose.*

*Proof.* Similar to Lemma 7.6, we will show that the head cannot return to $G_2$. To move from $G_1$ to $G_2$, the snake has to move into $G_1$ until it is either entirely in $G_1$, or only its tail outside of $G_1$. Hence, it must occupy at least $m$ vertices in $G_1$. But then the head also has to visit $a$, at which point it will grow to length $m + 2$. By Lemma 7.6, the snake will lose. □

We can now show that if the two components of $G - v$ have different sizes, then the apple placer has a winning strategy.

**Lemma 7.8.** *Let $G = (V, E)$ be a graph with $\kappa(G) = 1$ and let $v$ be a cut vertex of $G$. Let $G_1$ and $G_2$ be two different components of $G - v$. If $|V(G_1)| \neq |V(G_2)|$ then $G$ is not snake-winnable.*

*Proof.* First note that if either $|V(G_1)| = 1$ or $|V(G_2)| = 1$, then $G$ has a vertex of degree 1 and is not snake-winnable by Observation 2.5. Hence, we may assume that both $G_1$ and $G_2$ have at least two vertices.

Let $|V(G_1)| = m_1$ and $|V(G_2)| = m_2$ and suppose $m_1 \neq m_2$. Without loss of generality, we assume that $m_1 < m_2$. When the snake grows to length $m_1 - 1$ there must be some vertex $u_1$ in $G_1$ that is unoccupied. The apple placer chooses $u_1$ as the next apple location. When the snake eats the apple on $u_1$, it will grow to length $m_1$. Since $m_1 < m_2$, there must be some unoccupied vertex $u_2$ in $G_2$. The apple placer chooses $u_2$ as the next apple location.

When the snake eats the apple on $u_2$, it will either be entirely in $G_2$, or both the cut vertex $v$ and $u_2$ are occupied by the snake. Thus, at least two of the occupied vertices are in $G_2$. Since the snake now has length $m_1 + 1$, it follows that that must be some unoccupied vertex $v_1$ in $G_1$. The apple placer chooses $v_1$ as the next apple location. When the snake eats the apple on $v_1$, it will grow to length $m_1 + 2$ with its head in $G_1$. Furthermore, there must be an unoccupied vertex in $G_2$ and by Lemma 7.6, the snake will lose. □

Next, we show that if one of the two components is incomplete, then the snake will lose.

**Lemma 7.9.** *Let $G = (V, E)$ be a graph with $\kappa(G) = 1$ and let $v$ be a cut vertex of $G$. Let $G_1$ and $G_2$ be two different connected components of $G - v$ and suppose $G_1 + v$ and $G_2 + v$ are not both complete. Then $G$ is not snake-winnable.*

*Proof.* If $G_1$ and $G_2$ do not have the same number of vertices, then $G$ is not snake-winnable by Lemma 7.8. Hence, we may assume that $G_1$ and $G_2$ have the same number of vertices. Let $|V(G_1)| = |V(G_2)| = m$. If $m = 1$, then $G$ has a vertex of degree 1 and by Observation 2.5 $G$ is not snake-winnable. Hence, we may assume that $m \geq 2$.

Suppose that $G_1 + v$ and $G_2 + v$ are not both complete subgraphs of $G$. By symmetry, we assume $G_2 + v$ is incomplete.

When the snake reaches length $m - 1$, there must be some unoccupied vertex $u_2$ in $G_2$. The next apple is placed on $u_2$. When the snake eats the apple on $u_2$ and grows to length $m$, there will be at most $m - 2$ vertices in $G_1$ that are occupied. Hence, the apple placer can place the next apple on some $u_1$ in $G_1$, guaranteeing that the head of the snake is in $G_1$ when it grows to length $m + 1$. Furthermore, there must be an unoccupied vertex in $G_2$.

First, suppose that there is some unoccupied vertex $v_1$ in $G_1$. Then the apple placer places the next apple on $v_1$. Since the snake has length $m + 1$ and both its head and the apple are in $G_1$, by Lemma 7.7, the snake will lose.

Now suppose that there are no unoccupied vertices in $G_1$. Recall that we assumed $G_2+v$ is incomplete. If there is some vertex in $G_2$ that is not adjacent to $v$, then the apple placer places the next apple there. This means that if the snake moves to $v$ and then immediately eats the apple, then after eating the apple, the head graph is incomplete. Since the snake now has length $m+2$ and the circumference of $G$ is at most $m+1$, by Lemma 6.12, the snake will lose.

If the snake first moves to $v$ and then to a vertex in $G_2$ without the apple, then a vertex in $G_1$ will become unoccupied. Since the snake has length $m+1$ and both its head and the apple are in $G_2$, by Lemma 7.7 the snake will lose. ☐

**Theorem 7.10.** *Let $G=(V,E)$ have vertex-connectivity 1 and let $v$ be a cut vertex of $G$. Furthermore, let $G_1$ and $G_2$ be two different connected components of $G-v$. Then $G$ is snake-winnable if and only if $\omega(G-v)=2$, $|V(G_1)|=|V(G_2)|\geq 2$, and $G_1+v$ and $G_2+v$ are both complete.*

*Proof.* By Lemma 7.5, if $\omega(G-v)=2$, $|V(G_1)|=|V(G_2)|\geq 2$, and $G_1+v$ and $G_2+v$ are both complete, then $G$ is snake-winnable.

By Observation 7.2, $G$ is not snake-winnable if $\omega(G-v)>2$. If either $|V(G_1)|=1$ or $|V(G_2)|=1$, then $G$ had a vertex of degree 1 and is not snake-winnable by Observation 2.5. By Lemma 7.8, if $|V(G_1)|\neq|V(G_2)|$, then $G$ is not snake-winnable. By Lemma 7.9, $G$ is also not snake-winnable if either $G_1+v$ or $G_2+v$ is incomplete. ☐

For grid graphs, Theorem 7.10 gives us the following.

**Corollary 7.11.** *There are no snake-winnable grid graphs with vertex-connectivity 1.*

*Proof.* Let $G$ be a grid graph with $\kappa(G)=1$ with cut vertex $v$. Let $G_1$ and $G_2$ be two components of $G-v$. We know $G$ cannot have a clique with more than two vertices. But if $|V(G_1)|=|V(G_2)|\geq 2$ and $G_1+v$ and $G_2+v$ are both complete, the $V(G_1+v)$ and $V(G_2+v)$ are both a clique with at least 3 vertices. ☐

## 7.2 Snake-winnable graphs with vertex-connectivity 2

In Section 3 we saw a few examples of graphs with vertex-connectivity 2 that are non-Hamiltonian and snake-winnable, for example, the graph in Figure 3.7. In this section, will present two different types of non-Hamiltonian graphs that are snake-winnable.

Let $G$ be the graph in Figure 7.3, which can be constructed as follows. We first take a simple path $P=(p_1,\ldots,p_k)$ with $k\geq 3$. We then add $G_1$ and $G_2$, which are two copies of the complete graph on $m$ vertices. Finally, we make both $p_1$ and $p_k$ adjacent to all

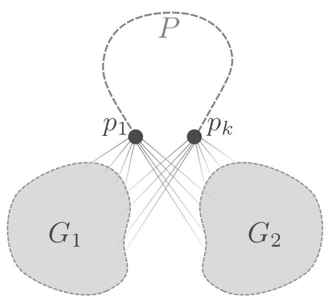



Figure 7.3: A snake-winnable graph with vertex-connectivity 2. The subgraphs $G_1$, $G_2$ are complete, $p_1$ and $p_k$ are adjacent to all the vertices in $V(G_1)\cup V(G_2)$.

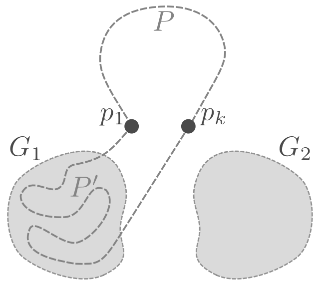



Figure 7.4: The snake moves along a cycle $(P, P')$ with $V(P') = V(G_1)$.

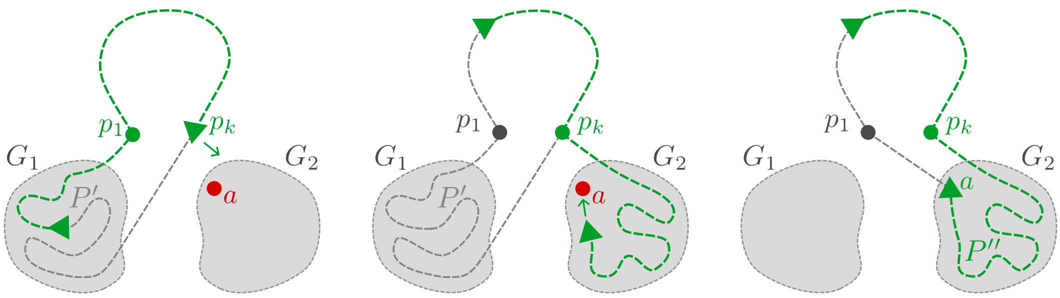
the vertices in both copies. Note that $S = \{p_1, p_k\}$ is a vertex cut and $G - S$ has three connected components. By Observation 7.2, $G$ is non-Hamiltonian. We now show that $G$ is snake-winnable.

**Lemma 7.12.** *The graph in Figure 7.3 is snake-winnable.*

*Proof.* We first show that up until the snake reaches length $k + m$, it can maintain the following. The snake always moves along a cycle $(P, P')$, with either $V(P') = V(G_1)$ or $V(P') = V(G_2)$, as depicted in Figure 7.4.

Suppose the snake has length $\ell < k + m$ and is moving along such a cycle, but the apple is placed on a vertex outside of the cycle. By symmetry, we may assume the snake is moving along a cycle $(P, P')$ with $V(P') = V(G_1)$, and the apple is placed on some vertex $a \in V(G_2)$. First, the head moves along the cycle until it reaches $p_k$ as depicted in Figure 7.5a. At this point, there will be $\ell - k < m$ occupied vertices in $G_1$. The head then visits the $m - 1$ vertices in $G_2$ that do not contain the apple. After doing so, the tail has moved $m - 1$ times, and thus the vertices in $G_1$ are all unoccupied, as depicted in Figure 7.5b. Furthermore, since $G_2$ is a complete graph, the head must be adjacent to the apple. The snake eats the apple, and since all vertices in $G_2$ are adjacent to $p_1$, it will be contained in some cycle $(P, P'')$ with $V(P'') = V(G_2)$, as depicted in Figure 7.5c.

By following this strategy, the snake can guarantee that once it reaches length $k+m$, it occupies exactly all the vertices in $P$ and all vertices in either $G_1$ or $G_2$. Furthermore, since it is on a cycle of exactly the same length as the snake, the head is adjacent to the tail. By symmetry, we may assume the snake occupies all the vertices in $V(P) \cup V(G_1)$. The head then repeatedly moves to the tail until it reaches $p_1$. Since $p_1$ is adjacent to all vertices in $G_2$, the head graph is now complete, and by Observation 6.14, the snake will win. □




(a) From $p_k$, the snake moves into $G_2$.

(b) The snake first visits all empty vertices in $G_2$.

(c) After eating the apple, the snake lies on $(P, P'')$.

Figure 7.5: Up until the snake reaches length $k + m$, it always moves along a cycle $(P, P')$, with either $V(P') = V(G_1)$ or $V(P') = V(G_2)$.

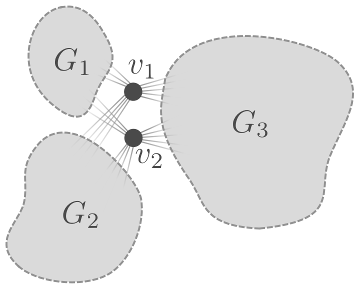



Figure 7.6: A snake-winnable graph with vertex-connectivity 2. The subgraphs $G_1$, $G_2$ and $G_3$ are all complete with $|V(G_3)| \geq |V(G_2)| \geq |V(G_1)|$. The vertices $v_1$ and $v_2$ are adjacent to all the vertices in $V(G_1) \cup V(G_2), \cup V(G_3)$.

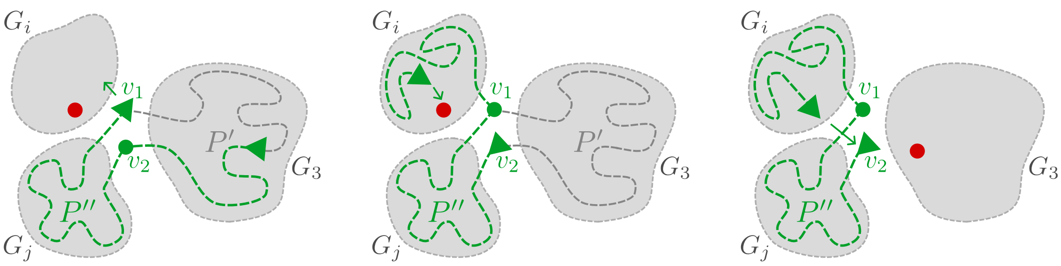
Recall that by Theorem 3.12, odd-sized bipartite graphs are snake-winnable if and only if they have a $\Theta(|V|-3,2,2)$ spanning subgraph. Note that for the graph in Figure 7.3, if $G_1$ and $G_2$ are $K_1$, then $G = \Theta(|V|-3,2,2)$.

In the strategy for Lemma 7.12, the snake has different options for the final cycle and final clique. It could either have $V(P) \cup V(G_1)$ on the final cycle and $V(G_2)$ as the final clique, or $V(P) \cup V(G_2)$ on the final cycle and $V(G_1)$ as the final clique. In the strategies we found for the graphs in Figure 3.7, Figure 3.15 and Figure 5.6, the snake also had several options for the final cycle and final clique. However, we will show that there are non-Hamiltonian snake-winnable graphs with vertex-connectivity 2 where the snake can always have the same final cycle and final clique.

Let $G$ be the graph in Figure 7.6, which can be constructed as follows. First, we take three complete graphs $G_1$, $G_2$, and $G_3$ with $|V(G_1)| = m_1$, $|V(G_2)| = m_2$, and $|V(G_3)| = m_3$. We have $m_3 \geq m_2 \geq m_1 \geq 2$ and $m_3 \leq m_1 + m_2 - 2$. We then add two vertices $v_1$ and $v_2$ and make both of these vertices adjacent to all vertices in $V(G_1) \cup V(G_2) \cup V(G_3)$. Note that $G$ has $\kappa(G) = 2$ with vertex cut $S = \{v_1, v_2\}$. Furthermore, since $\omega(G - S) = 3$, $G$ is not Hamiltonian.

**Lemma 7.13.** *Let $G$ be the graph in Figure 7.6. Then $G$ is snake-winnable and the snake can always end with $V(G_3)$ as a final clique.*

*Proof.* Since $G_3$ is complete, there is a $v_1v_2$-path $P$ that contains all the vertices in $G_3$. Note that $|P| \leq m_1 + m_2$. Up until the snake reaches length $m_1 + m_2 + 1$, it always moves around some cycle $C = (P, P')$ with $|C| = m_1 + m_2 + 1$ and either $V(P') \subseteq V(G_1)$ or $V(P') \subseteq V(G_2)$. This is possible since the snake can use the same strategy as in Lemma 7.12.




(a) Along $(P, P'')$, the snake moves to $v_1$ and then into $G_i$.

(b) The snake first visits all empty vertices in $G_i$.

(c) The snake is on a cycle and $V(G_3)$ is the final clique.

Figure 7.7: The snake can ensure $V(G_3)$ is the final clique.

When the snake reaches length $m_1 + m_2 + 1$, it will occupy all the vertices of its current cycle, which include $v_1, v_2$ and all the vertices in $G_3$. Hence, the next apple will either be placed in $G_1$ or $G_2$. Let $G_i \in \{G_1, G_2\}$ be the component that contains the apple and $G_j \in \{G_1, G_2\}$ the component that does not contain the apple.

The snake first moves to a cycle $(P, P'')$ where $V(P'') = V(G_j)$, as depicted in Figure 7.7a. When the snake reaches $v_1$, instead of continuing along $P$, the snake moves into $G_i$. There, it first visits all the $G_i$ that do not contain the apple, as depicted in Figure 7.7b. Finally, the snake eats the apple, after which it grows to a length $m_1 + m_2 + 2$. At this point, it will lie on a cycle that consists of all the vertices in $V(G_1) \cup V(G_2) \cup \{v_1, v_2\}$. Thus, the next apple must be placed in $G_3$. The snake then moves to the tail on $v_2$, as depicted in Figure 7.7c. From there, it can eat the apple in $G_3$ and enter the complete phase with $V(G_3)$ the final clique. □

## 7.3 Snake-winnable graphs with higher vertex-connectivity

We have seen non-Hamiltonian snake-winnable graphs with vertex-connectivity 1 and 2. Furthermore, there are Hamiltonian graphs with arbitrarily large vertex-connectivity, which are snake-winnable by Observation 2.3. The question remains whether there are any non-Hamiltonian snake-winnable graphs with higher vertex-connectivity. We will show that by connecting several complete graphs, we can construct a non-Hamiltonian snake-winnable graph with arbitrarily large vertex-connectivity.

**Theorem 7.14.** *There are non-Hamiltonian snake-winnable graphs with arbitrarily large vertex-connectivity.*

*Proof.* For a given $k$, we will construct a non-Hamiltonian graph $G$ with $\kappa(G) = k$ that is snake-winnable, which is depicted in Figure 7.8. First, we take $G_1, \ldots, G_{k+1}$, which are $k+1$ copies of the same complete graph on at least 2 vertices. We then add $k$ new vertices $\{v_1, \ldots, v_k\}$, and make each of these vertices adjacent to all the vertices in each of the $G_i$ for $i \in 1, \ldots, k+1$. Note that $G$ has vertex-connectivity $k$ with vertex cut $S = \{v_1, \ldots, v_k\}$. Furthermore, $\omega(G - S) = k + 1$, and thus $G$ is non-Hamiltonian by Observation 7.2.

We will now prove that $G$ is snake-winnable by showing it has the graph from Figure 7.3 as a spanning subgraph. Let $P_i$ be a path in $G_i$ that contains all the vertices of $G_i$. Note that such a path exists for each of the $G_i$ since they are complete. We then form the path $P = (v_1, P_2, v_2, P_3, v_3 \ldots, v_{k-1}, P_k, v_k)$, as depicted in Figure 7.9. The path $P$ contains all the vertices of $G$ except for $V(G_1)$ and $V(G_{k+1})$. Furthermore, the endpoints of $P$ are adjacent to all the vertices in $V(G_1) \cup V(G_{k+1})$. Hence, by combining $P$ with $G_1$, $G_{k+1}k$, and all edges between $\{v_1, v_k\}$ and $V(G_1) \cup V(G_{k+1})$, we obtain the graph from Figure 7.3. By Lemma 7.12, we know this graph is snake-winnable. Thus, $G$ has a snake-winnable spanning subgraph and it follows by Observation 2.4 that $G$ is snake-winnable. □

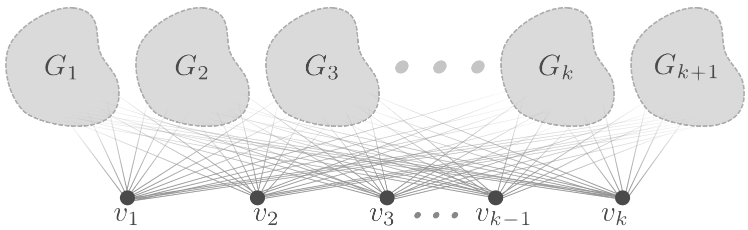



Figure 7.8: A non-Hamiltonian snake-winnable graph with vertex-connectivity k.

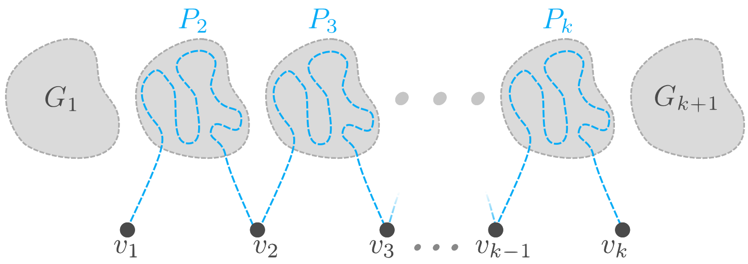



Figure 7.9: The path $P = (v_1, P_2, v_2, P_3, v_3 \ldots, v_{k-1}, P_k, v_k)$.

While the graph in Figure 7.8 has a snake-winnable spanning subgraph with lower vertex-connectivity, this is not the case for the graph in Figure 7.3. Since the latter has vertex-connectivity 2, by Theorem 7.10, the only snake-winnable graph of lower vertex-connectivity is the one in Figure 7.1. Hence, because of the path $P$, the graph in Figure 7.3 can never have a snake-winnable spanning subgraph of vertex-connectivity 1. We can thus consider the graph in Figure 7.3 to be a minimal snake-winnable graph with respect to the vertex-connectivity. Whether such minimal graphs also exist for higher vertex-connectivities remains an open question.

# 8. Conclusion

Based on the classic computer game of Snake, we defined the game of Snake on a graph. Originally played on a grid, Snake can easily be translated to a game on a grid graph, which can then be further generalized to any connected simple graph. In the game of Snake on a graph, we disregard human factors like reaction time and focus solely on the combinatorial question of whether a graph is winnable. This makes the game of Snake on a graph a natural addition to the field of combinatorial games on graphs. Central to this game is the question of which non-Hamiltonian graphs are snake-winnable. Unfortunately, fully resolving this question remains an open problem.

For odd-sized bipartite graphs and graphs with vertex-connectivity 1, we were able to completely characterize snake-winnable graphs (Theorems 3.12 and 7.10). More generally, we showed that every non-Hamiltonian snake-winnable graph has a girth of at most 6 (Theorem 5.11). Furthermore, we determined that any non-Hamiltonian snake-winnable graph $G$ must have a cycle and a clique that are completely disjoint and together contain all the vertices of $G$ (Theorem 6.18). While both the condition on the girth, as well as the cycle and clique structure, are necessary for non-Hamiltonian graphs to be snake-winnable, they are not sufficient.

The snake problem is NP-hard, even when restricted to grid graphs (Theorem 4.1). Whether it can be solved in polynomial time on solid grid graphs remains an open question. Given the numerous possible apple placements and positions of the snake, it is unclear whether the snake problem is even in NP, as it is not obvious how a snake strategy could be described compactly.

## Acknowledgements

As I wrap up my thesis, I am starting realize how much I will miss daydreaming about snakes and apples. Working on this topic has been a genuinely enjoyable experience, and I owe that to the support of many.

First and foremost, I want to thank my supervisors Bodo Manthey and Alexander Skopalik. I am not exaggerating when I say that this thesis would not have existed without them. Even when my ideas were little more than cryptic scribbles on a whiteboard, they took the time to decipher them and helped me shape them into comprehensible proofs.

I truly appreciate all the DMMP members who allowed me to share my research and provided invaluable input. I am looking forward to spending the next four years in your presence and learning from your expertise.

Before I could even begin this final assignment, I spent many hours in classrooms learning from many great teachers. I want to thank everyone involved in the Applied Mathematics program for helping me grow as a mathematician and introducing me to so many fascinating subjects. In particular, I want to express my gratitude to Lilian Spijker, our study advisor. While studying can be stressful and hectic at times, she helped me navigate it with an unexpected level of ease and comfort.

None of this would have been possible without my parents. Thank you for your patience, your understanding, and for always believing in me. It could not have been easy to watch me grow through trial and error, but I will forever be grateful that you gave me the time and space to do so. I also want to thank both of my sisters for being incredible role models, each in their own unique way. Whether you give me big shoes to fill or easy footsteps to follow, I am glad to have you by my side.

I feel incredibly fortunate to have such an amazing group of friends who always seem to know exactly when I need a distraction. Whether it is a quick coffee break, a shared meal, or a late-night hangout, I cherish every moment we spend together and will continue to hold our friendship close to my heart.

A very special thanks goes out to Nova, my little furry friend, who made sure I took breaks by claiming my keyboard for her naps.

Finally, I am extremely lucky to have the unwavering support of my partner. For a physicist to listen to so many of my mathematical musings can only mean one of two things: either you love me even more than I imagined, or you secretly enjoy math. Both would make me very happy, but one significantly more than the other.

Thank you all for being part of this journey, for inspiring and guiding me, for lifting up my spirits, and for making each step of the way memorable.

# References


[1] J.A. Bondy and U.S.R. Murty. *Graph Theory with Applications*. American Elsevier Publishing Company, 1976. ISBN: 9780444194510.

[2] Shao Dong Chen, Hong Shen, and Rodney Topor. "An efficient algorithm for constructing Hamiltonian paths in meshes". In: *Parallel Comput.* 28.9 (2002), pp. 1293–1305. ISSN: 0167-8191,1872-7336. DOI: `10.1016/S0167-8191(02)00135-7`.

[3] Marzio De Biasi and Tim Ophelders. "The complexity of snake and undirected NCL variants". In: *Theoret. Comput. Sci.* 748 (2018), pp. 55–65. ISSN: 0304-3975,1879-2294. DOI: `10.1016/j.tcs.2017.10.031`.

[4] Erik D. Demaine. "Playing Games with Algorithms: Algorithmic Combinatorial Game Theory". In: *CoRR* cs.CC/0106019 (2001). URL: `https://arxiv.org/abs/cs/0106019`.

[5] Alon Itai, Christos H. Papadimitriou, and Jayme Luiz Szwarcfiter. "Hamilton paths in grid graphs". In: *SIAM J. Comput.* 11.4 (1982), pp. 676–686. ISSN: 0097-5397. DOI: `10.1137/0211056`.

[6] MobyGames. *Blockade*. `https://www.mobygames.com/game/80291/blockade/`. Accessed: 2024-08-16.

[7] MobyGames. *Nokia 6110 Included Games*. `https://www.mobygames.com/game/206796/nokia-6110-included-games/`. Accessed: 2024-08-16.

[8] MobyGames. *Rock-Ola's Nibbler*. `https://www.mobygames.com/game/59918/rock-olas-nibbler/`. Accessed: 2024-08-16.

[9] MobyGames. *Snake Byte*. `https://www.mobygames.com/game/42270/snake-byte/`. Accessed: 2024-08-16.

[10] Christos H. Papadimitriou. *Computational Complexity*. Theoretical computer science. Addison-Wesley, 1994. ISBN: 9780201530827.

[11] Christopher Umans and Willia Lenhart. "Hamiltonian cycles in solid grid graphs". In: *Proceedings 38th Annual Symposium on Foundations of Computer Science*. 1997, pp. 496–505. DOI: `10.1109/SFCS.1997.646138`.