Frog jumping problem on simple graphs

Veselovac, Matej

J.J. Strossmayer University of Osijek, Croatia
Department of Mathematics
Graduate university study programme in mathematics and computer science

Matej Veselovac

Frog jumping problem on simple graphs

Final thesis

Osijek, 2022.

J.J. Strossmayer University of Osijek, Croatia
Department of Mathematics
Graduate university study programme in mathematics and computer science

Matej Veselovac

Frog jumping problem on simple graphs

Final thesis

Supervisor: Associate professor Snježana Majstorović

Osijek, 2022.

Frog jumping problem on simple graphs

Summary

Frog jumping problem is played on a simple connected graph. Initially, one frog is placed on each vertex of a graph and the goal is to move all frogs to a single vertex. If there are m frogs on a vertex u, then it is allowed to move them on a vertex v if and only if u and v are connected by a shortest path consisted of m edges.

We solve the problem for path graphs, star graphs, starfish graphs, and a subset of dandelion graphs. We conjecture that all vertices of a sufficiently large complete binary tree are solvable. The problem is a generalization of the "Frog jumping" problem by G. Hamilton.

Keywords

Graph, simple graph, connected graph, path graph, star graph, starfish graph, dandelion graph, binary tree, spanning tree

Problem skakanja žaba na jednostavnim grafovima

Sažetak

Problem skakanja žaba se postavlja na jednostavnom povezanom grafu. Na početku se postavlja jedna žaba na svaki vrh, s ciljem da se sve žabe u konačnici sretnu u istom vrhu. Pomicanje svih x žaba s jednog vrha na drugi je dozvoljeno ako i samo ako su izabrani vrhovi neprazni i povezani putem koji se sastoji od x različitih bridova.

Rješavamo problem na putevima, zvijezdama, zvjezdača grafovima, te na podskupu maslačak grafova. Nagađamo da su svi vrhovi na svim dovoljno velikim potpunim binarnim stablima rješivi. Ovaj problem je generalizacija problema "Frog jumping" od G. Hamilton.

Ključne riječi

Graf, jednostavan graf, povezan graf, put graf, zvijezda graf, zvjezdača graf, maslačak graf, binarno stablo, razapinjuće stablo

Contents

Introduction **1**

1 Frog jumping problem **1**
1.1 Definitions and notation . . . 1
1.2 Algorithmic approach and complexity . . . 3

2 Simple tree graph classes **4**
2.1 Solving the problem on Path P_n graphs . . . 5
2.2 Solving the problem on Star S_m graphs . . . 6
2.3 Solving the problem on Starfish $S_{m,n}$ graphs . . . 6
2.4 Solving the problem on generalized $S_{(a_m)}$ graphs . . . 8

3 Dandelion $D_{m,n}$ graphs **11**
3.1 Solving vertices of Star $S'_m \subset D_{m,n}$ subgraph . . . 12
3.2 Solving vertices of Path $P'_n \subset D_{m,n}$ subgraph . . . 15

4 Unsolvable tree graphs **21**

5 Complete binary tree graphs **22**

6 Results **23**

References **24**

Introduction

Frog jumping problem is a generalization of the "Frog jumping". The original problem was featured in a Youtube video by Numberphile, starring G. Hamilton (see [4]). We contacted G. Hamilton, which led to online presentations of the generalization (see [2, 3]).

1 Frog jumping problem

Given a simple connected graph, the goal of the problem is to have all frogs meet at a single vertex. Initially, every vertex has one frog. All m frogs placed at vertex u can jump to vertex v if and only if both vertices contain at least one frog and there is a shortest path between u and v consisting of m edges. If a sequence of jumps exists such that all frogs end up on a single vertex, then the meeting can be successfully held. Such a vertex is called "solvable", a "lazy toad" or "lazy" because the frog in that vertex has never jumped.

1.1 Definitions and notation

We introduce the following definitions to formalize the problem.

Definition 1.1. *Let G be a simple connected graph with vertex set V.*

A **problem state** *is a vertex-weighted graph G^f where $f : V \to \mathbb{Z}^*$ assigns a nonnegative integer weight to each vertex such that $\sum_{v \in V} f(v) = |V|$.*

An **initial problem state** *is a problem state G^{f_0} such that $\forall v \in V, f_0(v) = 1$.*

A **solved problem state** *in vertex $v \in V$ is a problem state G^{f_x} such that $f_x(v) = |V|$. In this case, vertex v is called a* **solved vertex**.

In the above definition, $f(v)$ stands for the number of frogs placed on a vertex $v \in V$.

Definition 1.2. *Let G^f, G^h be two problem states on graph G with vertices in V, edges in E. We say that G^h is* **reachable** *from G^f and write $G^f \to G^h$ if there exists vertices $v, w \in V$ such that:*

1. *$\forall u \in V \setminus \{v, w\}, f(u) = h(u)$.*
2. *$h(v) = 0$, $h(w) = f(v) + f(w)$ and $f(v) \neq 0, f(w) \neq 0$.*
3. *There exists a path from v to w with exactly $f(v)$ unique edges $e_1, \dots, e_{f(v)} \in E$. We write $f(v) \in d(v, w)$, where d is the set of edge distances along any path from v to w.*

We write $G^h = G^f(v \to w)$ or $G^h = G^f(v \xrightarrow{f(v)} w)$ to emphasise the value of $f(v)$.

If G is acyclic, then $|d(v, w)| = 1$. For example, if G is a tree graph.

Definition 1.3. *Let G be a simple connected graph with vertices in V.*

Let $G^{f_0} \to G^{f_1} \to \cdots \to G^{f_x}$ be a sequence of reachable problem states ending in a solved problem state in vertex $v \in V$. We call these states **solvable states**. *If G^{f_0} is an initial problem state, then v is a* **solvable vertex** *and the sequence is a* **solution sequence**.

The question we can now ask is: "Given a simple connected graph G, which vertices are solvable?". When we solve a graph G, we represent solution sequences with "vertex to vertex" transitions.

Denote transitions between reachable states over some vertices v, w as follows.

- Write $(v \to w)$ for a simple transition. To emphasize the value, write $(v \xrightarrow{f(v)} w)$.
- A chained transition sequence $(v_1 \to v_2)(v_2 \to v_3)(v_3 \to v_4)\dots$ is written in short notation as $(v_1 \to v_2 \to v_3 \to v_4 \to \dots)$.
- A concentrated transition sequence $(v_1 \to v_0)(v_2 \to v_0)(v_3 \to v_0)\dots$ is written in short notation as $(\{v_1, v_2, v_3 \dots\} \to v_0)$.

For example, given a simple connected graph $G \simeq D_{2,2}$ with vertices $V = \{1, 2, 3, 4, 5\}$ and edges $E = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{4, 5\}\}$, we determine which vertices are solvable.

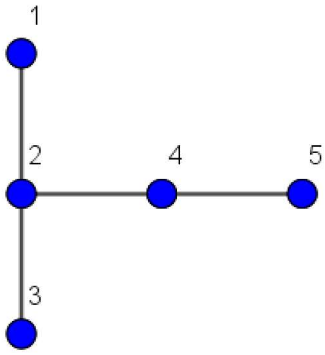

Image 1: Example graph G isomorphic to a dandelion $D_{2,2}$ graph.

The vertices 1 and 3 are symmetric, so if one is solvable, so is the other. To show that vertex 1 is solvable, find a solution sequence that starts with the initial problem state G^{f_0} and ends in a solved problem state $G^{f_x^1}$ whose solved vertex is 1. One such solution sequence is:

$$G^{f_x^1} = G^{f_0}(2 \xrightarrow{1} 3)(3 \xrightarrow{2} 1)(5 \xrightarrow{1} 4)(4 \xrightarrow{2} 1) = G^{f_0}(2 \xrightarrow{1} 3 \xrightarrow{2} 1)(5 \xrightarrow{1} 4 \xrightarrow{2} 1).$$

Similarly, one possible solution sequence for vertex 2 is:

$$G^{f_x^2} = G^{f_0}(1 \xrightarrow{1} 2)(3 \xrightarrow{1} 2)(4 \xrightarrow{1} 5)(5 \xrightarrow{2} 2) = G^{f_0}(\{1, 3\} \xrightarrow{1+1} 2)(4 \xrightarrow{1} 5 \xrightarrow{2} 2).$$

Last but not least, a solution sequence for vertex 5 is:

$$G^{f_x^5} = G^{f_0}(2 \xrightarrow{1} 1)(1 \xrightarrow{2} 3)(3 \xrightarrow{3} 5)(4 \xrightarrow{1} 5) = G^{f_0}(2 \xrightarrow{1} 1 \xrightarrow{2} 3 \xrightarrow{3} 5)(4 \xrightarrow{1} 5).$$

On the other hand, the problem is not solvable in vertex 4. Since the graph G has only five vertices, this can be shown by exhausting all sequences of reachable problem states.

Alternatively, let $v, w \in V$ and observe that the distance from vertex 4 to any other vertex is at most 2. Therefore, a solution sequence can contain only $(v \xrightarrow{1} w), (v \xrightarrow{2} 4)$ transitions. Since 1 and 3 are at a distance of 2, they require one $(v \xrightarrow{2} 4)$ transition. But, two $(v \xrightarrow{2} 4)$ transitions are impossible as their only neighbor is shared.

1.2 Algorithmic approach and complexity

By definition, we can solve the problem recursively as follows. The graph G with $n = |V|$ vertices is solvable in vertex $v \in V$ if and only if there exists a partition $\{V_1, V_2, \ldots, V_k\}$ of vertices $V \setminus \{v\}$ and vertices $v_i^* \in V_i, i = 1, \ldots, k$ with the following properties:

1. $\forall i$, $|V_i| \in d(v, v_i^*)$.

2. $\forall i$, problem state G^{f_i}, $f_i(v) = \begin{cases} 0, & v \notin V_i \\ 1, & v \in V_i \setminus \{v_i^*\} \\ |V| - |V_i|, & v = v_i^* \end{cases}$, is solvable in $v_i^* \in V_i$.

We can apply this recursively to partition parts, parts of partition parts, and so on. Essentially, we go backwards from the last to the first transitions.

That is, we start by computing all the distance sets d. Then the problem is to construct a directed spanning tree. We begin at the target vertex v and proceed to follow the appropriate distances. In the worst case, we exhaust all relevant partitions.

Problems involving partitioning graphs into subgraphs of certain types, such as triangles or isomorphic subgraphs, tend to be exponential (see [1]). Therefore, it is likely that there is no efficient algorithm for general graphs.

For example, we can solve all tree graphs with less than 15 vertices. The following table shows the number of graphs, where the rows and columns denote the number of vertices and unsolvable vertices, respectively.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0													
2	1	0	0												
3	1	0	0	0											
4	1	0	0	1	0										
5	1	1	0	0	1	0									
6	2	1	1	0	0	2	0								
7	3	2	1	1	0	2	2	0							
8	6	5	4	1	0	1	1	3	2						
9	12	8	6	8	1	0	3	0	6	3					
10	26	23	13	15	9	0	2	0	1	12	5				
11	59	44	42	24	14	8	3	5	3	3	19	11			
12	141	115	82	57	49	21	6	8	6	8	11	28	19		
13	348	270	201	132	95	59	33	16	19	6	9	11	61	41	
14	911	647	464	307	218	149	96	47	35	29	22	21	16	115	82

Table 1: Results on the solvability of the vertices of tree graphs with $n < 15$ vertices.

In the following section, we find solution sequences for specific classes of graphs. These strategies can be applied to similar subgraphs in other graphs.

2 Simple tree graph classes

We attempt to solve the problem on four simple classes of simple connected tree graphs. However, some subcases remain unsolved.

Definition 2.1 (see [12]). *A simple graph, also called a strict graph, is an unweighted, undirected graph that contains no graph loops or multiple edges.*

We examine the problem on path P_n, star S_m, starfish $S_{m,n}$ and dandelion $D_{m,n}$ graphs.

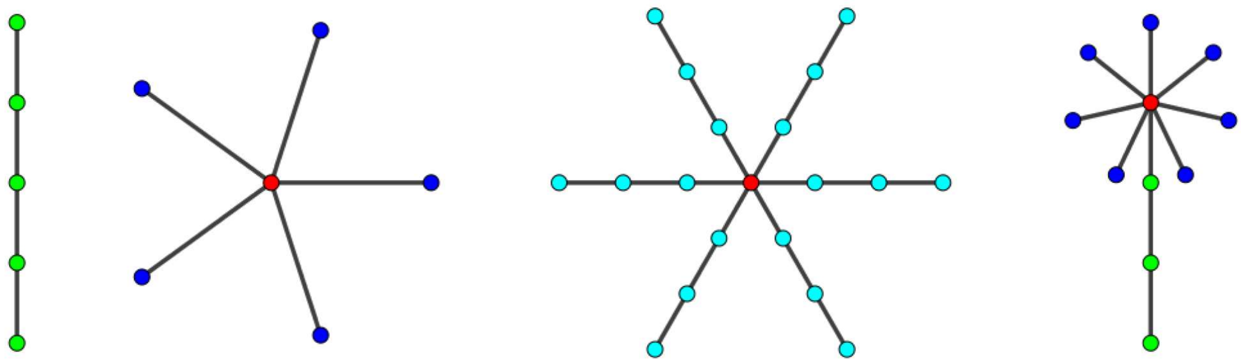

Image 2: Illustrations of path P_5, star S_5, starfish $S_{6,3}$, dandelion $D_{7,3}$.

These graph classes can be described as:

1. Path P_n is a tree graph with 2 leaf vertices and $n-2$ vertices of degree 2.
2. Star S_m is a tree graph with m leaf vertices connected to a center vertex of degree m. Alternatively, this is a complete bipartite graph $K_{1,m}$.
3. Starfish $S_{m,n}$ consists of m paths P_n connected to a center vertex of degree m.
4. Dandelion $S_{m,n}$ consists of a star S_m connected to a path P_n via its center vertex.

Note, for example, that $S_m = S_{m,1}$, $P_{2n+1} = S_{2,n}$ and $D_{m,1} = S_{m+1}$. All four classes are classes of tree graphs. A tree graph is a simple connected graph that does not contain cycles (is acyclic).

A similar definition of a dandelion $D_{m,n}$ graph comes from M. Krnc and R. Škrekovski (see [5]). We have not found any reference that uses the name "starfish" for $S_{m,n}$.

A generalization involving all four classes can be defined as $S_{(a_m)}$, where (a_m) is a finite sequence of nonnegative integers. Then $S_{(a_m)}$ consists of paths $P_{a_1}, \ldots, P_{a_m}$ joined in a vertex of degree m. For example, $S_{m,n} = S_{(\underbrace{n, \ldots, n}_{m})}$. These graphs are also called "starlike".

A hard generalization would be to consider all tree graphs (see [8]). Individual examples can be found in a `mathpickle.com` article by G. Hamilton (see [2]).

In addition to these four classes of trees, we also observe the problem on complete binary tree graphs T_h of height h, consisting of $|V(T_h)| = 2^{h+1} - 1$ vertices.

2.1 Solving the problem on Path P_n graphs

Path P_n graphs were solved by G. Hamilton in 2017 (see [4]). That is, this case is equivalent to the original "Frog jumping" or "Lazy toad" problem.

It is not difficult to show that all vertices of a path P_n are solvable. It is possible to use a simple strategy that alternates "left" and "right" jumps.

Lemma 2.2. *Leaf vertices of a path P_n are solvable for every $n \in \mathbb{N}$.*

Proof:
If n is odd, let vertices be $V = \{-k, -k+1, \ldots, -1, 0, 1, \ldots, k-1, k\}$, $k = \frac{n-1}{2}$. Then we have the following solution sequences:

$$\begin{aligned} G^{f_x^{-k}} &= G^{f_0}(0 \xrightarrow{1} +1 \xrightarrow{2} -1 \xrightarrow{3} +2 \xrightarrow{4} -2 \xrightarrow{5} \ldots \xrightarrow{n-1} -k). \\ G^{f_x^{k}} &= G^{f_0}(0 \xrightarrow{1} -1 \xrightarrow{2} +1 \xrightarrow{3} -2 \xrightarrow{4} +2 \xrightarrow{5} \ldots \xrightarrow{n-1} +k). \end{aligned}$$

Otherwise, if n is even, let the vertices be $V = \{-k, -k+1, \ldots, -1, 1, \ldots, k-1, k\}$, $k = \frac{n}{2}$. Then we have the following solution sequences:

$$\begin{aligned} G^{f_x^{-k}} &= G^{f_0}(+1 \xrightarrow{1} -1 \xrightarrow{2} +2 \xrightarrow{3} -2 \xrightarrow{4} +3 \xrightarrow{5} \ldots \xrightarrow{n-1} -k). \\ G^{f_x^{k}} &= G^{f_0}(-1 \xrightarrow{1} +1 \xrightarrow{2} -2 \xrightarrow{3} +2 \xrightarrow{4} -3 \xrightarrow{5} \ldots \xrightarrow{n-1} +k). \end{aligned}$$

That is, starting from the center, the strategy is to alternate left and right jumps. □

Theorem 2.3. *All vertices of a path P_n are solvable for any $n \in \mathbb{N}$.*

Proof:
Let $V = \{1, 2, 3, \ldots, n\}$ be vertices of a path P_n. Note that for each vertex $a \in V$, path subgraphs $P_a, P_{n-a+1} \subseteq P_n$ induced by the vertices $\{1, 2, 3, \ldots, a\}$ and $\{a, a+1, a+2, \ldots, n\}$ are solvable in the common vertex a, according to Lemma 2.2.

Therefore, P_n is solvable in a. Since $a \in V$ is arbitrary, every vertex of P_n is solvable. □

Since adding more edges to a graph can only increase the number of reachable problem states, we have the following corollary.

Corollary 2.4. *All vertices of a graph G containing a Hamiltonian path are solvable.*

The above corollary implies that, for example, all vertices of a cycle C_n and all vertices of a complete graph K_n are solvable.

More generally, note that graphs whose spanning trees are solvable in every vertex are solvable in every vertex.

A possible generalization would be to connect a leaf vertex to each vertex of a path P_n. Then, we get a "centipede" c_n graph with $|V(c_n)| = 2n$ vertices (see [11]). It is not difficult to show that all vertices of every centipede c_n graph are solvable. Indeed, a similar strategy as in the previous theorem can be used.

2.2 Solving the problem on Star S_m graphs

We can assume $m \geq 3$. Otherwise, we have $S_m \simeq P_2, P_3$ for $m = 1, 2$, respectively.

Theorem 2.5. *In a star $S_m, m \geq 3$, only the center vertex of degree m is solvable.*

Proof:
Solving a star S_m with vertices $V = \{0, 1, 2, \ldots, m\}$ in the center vertex 0 is simple:

$$G^{f_x} = G^{f_0}(\{1, 2, 3, \ldots, m\} \to 0).$$

Otherwise, note that the distance from one leaf vertex $u \in V$ to every other leaf vertex $v \in V$ is equal to the diameter 2 of the graph. But the leaf vertices are not connected to each other. Therefore, we need $m - 1$ transitions $(v \xrightarrow{2} u)$. This is a contradiction, because $2m - 2 > m$. As a result, leaf vertices are not solvable. □

Now we are ready to move on to the starfish $S_{m,n}$ graphs.

2.3 Solving the problem on Starfish $S_{m,n}$ graphs

A few examples and strategies can be found in a `mathpickle.com` article by G. Hamilton (see [3]). Given a starfish $S_{m,n}$, we can assume $m \geq 3$ and $n \geq 2$. Let v_0 be the center vertex of degree m. We show that all vertices are solvable.

Lemma 2.6. *In a starfish $S_{m,n}$, the center vertex of degree m is solvable.*

Proof:
Consider m path subgraphs P_{n+1} of $S_{m,n}$ that do not share any vertices except the vertex v_0. According to Lemma 2.2, every such subgraph is solvable in v_0. Consequently, a starfish $S_{m,n}$ is solvable in the center vertex v_0. □

It remains to solve non-center vertices. Due to symmetry, we only need to consider finding up to n solution sequences for a given starfish $S_{m,n}$.

Consider m disjoint path subgraphs $P_n^i, i = 1, 2, \ldots, m$ of $S_{m,n}$ that are connected to the center vertex v_0. Such subgraph is referred to as a "tentacle".

Let $v_d^i \in V(P_n^i)$ be a vertex at $d \in \{1, 2, \ldots, n-1\}$ edges distance from the vertex v_0.

Lemma 2.7. *In a starfish $S_{m,n}, n \geq 2$, vertices of degree 2 are solvable.*

Proof:
For $m \in \{1, 2\}$ see the Theorem 2.3. Otherwise, proceed as follows:

1. Solve tentacles $P_n^i, i = 1, 2, \ldots, m-1$ in the vertex v_d^i.
2. Apply transitions $(v_d^i \xrightarrow{n} v_{n-d}^m)$.
3. Solve subgraph $(P_n^m + v_0)$ in v_{n-d}^m.

Steps 1. and 3. are solvable due to the Theorem 2.3. □

To solve the leaf vertices, we use two strategies. For $n = 2$, we solve every other tentacle using its neighbors. Otherwise, we use a similar strategy to solve three tentacles at a time.

Lemma 2.8. *In a starfish $S_{m,n}, n \geq 2$, leaf vertices are solvable.*

Proof:
For $m \in \{1, 2\}$ see the Theorem 2.3. Otherwise, it is sufficient to show that the vertex v_n^1 is solvable. Let $I(x) = x$ except $I(1) = m - 1 + (m \bmod 2)$. If $n = 2$, proceed as follows:

1. Apply transitions $(v_2^{2k} \xrightarrow{1} v_1^{2k} \xrightarrow{2} v_1^{I(2k-1)} \xrightarrow{3} v_2^{2k+1} \xrightarrow{4} v_2^1)$ for all $k = 1, 2, \ldots, \lfloor \frac{m-1}{2} \rfloor$.
2. If m is odd, apply $(v_1^1 \xrightarrow{1} v_0 \xrightarrow{2} v_2^1)$. Otherwise, apply $(v_0 \xrightarrow{1} v_1^m \xrightarrow{2} v_1^1 \xrightarrow{3} v_2^m \xrightarrow{4} v_2^1)$.

If $n \geq 3$, for all $k = 1, 2, \ldots, \lfloor \frac{m-1}{3} \rfloor$ proceed as follows:

1. Solve tentacle P_n^{3k-1} in v_{n-1}^{3k-1} and apply $(v_{n-1}^{3k-1} \xrightarrow{n} v_1^{3k} \xrightarrow{n+1} v_n^1)$.
2. Solve $(P_n^{3k} - v_1^{3k})$ in v_{n-1}^{3k}, P_n^{3k+1} in v_1^{3k+1} and apply $(v_1^{3k+1} \xrightarrow{n} v_{n-1}^{3k} \xrightarrow{2n-1} v_n^1)$.

The steps are solvable due to the Theorem 2.3. Consequently, we are left with $0, 1$ or 2 additional tentacles beside the P_n^1 tentacle. In the case of 0 or 1, the remaining tentacles form a path subgraph to which we apply the Theorem 2.3. Otherwise, we are left with 2 of additional tentacles. In this case, proceed as follows:

1. Solve subgraph $(P_n^1 + v_0 + v_1^{m-1})$ in vertex v_n^1.
2. Solve $(P_n^{m-1} - v_1^{m-1})$ in v_{n-1}^{m-1}, P^m in v_1^m and apply $(v_1^m \xrightarrow{n} v_{n-1}^{m-1} \xrightarrow{2n-1} v_n^1)$.

The steps are solvable due to the Theorem 2.3. $\square$

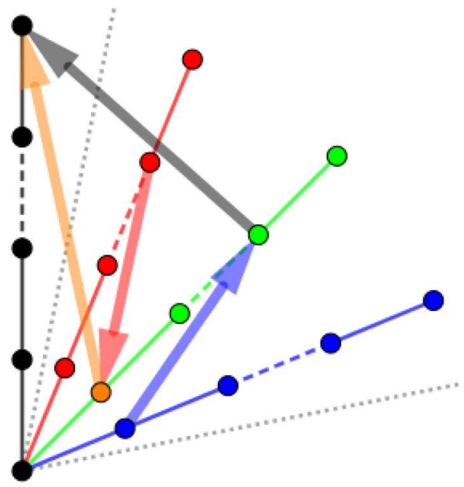

Image 3: Reduction of 3 tentacles by moving all their frogs to the v_n^1 leaf.

Theorem 2.9. *In a starfish $S_{m,n}, n \geq 2$, all vertices are solvable.*

Theorem 2.9 is a consequence of Lemmas 2.6, 2.7, 2.8. A natural generalization is to consider tentacles of different lengths. This is solved in the following subsection.

2.4 Solving the problem on generalized $S_{(a_m)}$ graphs

Let $S_{(a_m)}$ be a graph consisting of paths $P_{a_1}, P_{a_2}, \ldots, P_{a_m}$ called "tentacles". They are connected in the vertex v_0 of degree m, where (a_m) is a finite sequence of positive integers. Due to the same arguments as in Lemma 2.6, these graphs are solvable in vertex v_0.

The "balanced" case of $S_{(a_m)}$, where $a_1 = a_2 = \cdots = a_m = n$, is solvable in every vertex according to the Theorem 2.9.

It is not hard to see that the "unbalanced" case, where $a_1 \neq a_2 \neq \ldots \neq a_m$, is also solvable in every vertex. We formulate this in the following theorem.

Theorem 2.10. *Generalized $S_{(a_m)}$ graph is solvable in every vertex if $a_1 \neq a_2 \neq \ldots \neq a_m$.*

Proof:
The proof is inductive. The assumption is that $S_{(a_m)}$ is solvable in every vertex. We need to show that $S_{(a_{m+1})}$ is solvable in every vertex for every $a_{m+1} \in \mathbb{N}$.

Take $a_{m+1} > a_m > a_{m-1} > \cdots > a_1 \geq 1$ without loss of generality. If $(m+1) \in \{1, 2\}$, apply the Theorem 2.3. Otherwise, we can use the following result.

If $a_i > a_j$, all frogs from the tentacle P_{a_i} can be moved to any vertex of the tentacle P_{a_j} due to the steps used in Lemma 2.7. We refer to this strategy as "reducing the tentacle P_{a_i} into a vertex of the tentacle P_{a_j}."

The graph $G = S_{(a_{m+1})}$ can be decomposed into two disjoint subgraphs $G_1 = S_{(a_m)}$ and $G_2 \simeq P_{a_{m+1}}$. To solve G in the vertex $v \in V(G_1)$, first solve G_1 in v using the assumption. If $v \neq v_0$, reduce the tentacle $G_2 \simeq P_{a_{m+1}}$. Otherwise, apply the Theorem 2.3 to $(G_1 + v_0)$ in $v = v_0$. Remains to solve G in vertices $V(G_2) = \{v_1, v_2, \ldots, v_{a_{m+1}}\}$, where v_d is at distance d from vertex v_0. Let $v_d \in V(G_2)$ be the vertex we want to solve.

Start by reducing all possible tentacles $P_{a_i} \in G_1$ into the vertex v_d. If we have reduced all tentacles, we are done due to the Theorem 2.3 on $(G_2 + v_0)$ in v_d. Otherwise, we are left with tentacles $P_{c_1}, \ldots, P_{c_r}$ with $d \geq c_1 > c_2 > \cdots > c_r \geq 1$. When $r = 1$ we are done due to the Theorem 2.3 on $(G_2 + v_0 + P_{c_1})$ in v_d. Therefore, $r \geq 2$. Let k be the largest index such that $s_k = \sum_{i=1}^{k} c_i \leq d$. If $k = r$, apply the following procedure:

1. Reduce the tentacles $P_{c_1}, \ldots, P_{c_{k-1}}$ to the vertex $v_1^{(c_k)}$ of the shortest tentacle P_{c_k}.
2. Solve $(P_{c_k} + v_0 + v_1 + \cdots + v_{d-s_k})$ in $v_1^{(c_k)}$ using the Theorem 2.3 and apply $(v_1^{(c_k)} \xrightarrow{d+1} v_d)$.
3. Solve the remaining part of G_2 using the Theorem 2.3.

If $k < r$, then $s_{k+1} = d + t, t \in \mathbb{N}$, $c_{k+1} \geq t$, and apply the following procedure first:

1. Reduce the tentacles $P_{c_1}, \ldots, P_{c_k}$ to the vertex $v_t^{(c_{k+1})}$ of the shorter tentacle $P_{c_{k+1}}$.
2. Solve $P_{c_{k+1}}$ in $v_t^{(c_{k+1})}$ and apply the transition $(v_t^{(c_{k+1})} \xrightarrow{d+t} v_d)$.
3. Repeat steps 1. and 2. for the remaining tentacles until $k = r$.

This completes the inductive argument. □

It remains to consider the case that $S_{(a_n)}$ is neither "balanced" nor "unbalanced". It is possible to prove a stronger alternative to the previous theorem.

We will need two new lemmas. First, we need to introduce "waterlogged" stars $S^*_{m,n}, n \geq 2$. Let v_0 be the central vertex of a star $S_{m,n}$ graph. A waterlogged star $S^*_{m,n}$ is solvable in a vertex $v \neq v_0$ if the problem state G^{f_*},

$$f_*(u) = \begin{cases} 1, & u = v_0 \\ |V| - 1, & u = v \\ 0, & \text{else} \end{cases},$$

on a star $S_{m,n}$, is reachable from the initial problem state. Note that G^{f_*} is the same as the solved problem state, except that we are ignoring the frog on the central vertex v_0.

Lemma 2.11. *In a waterlogged star $S^*_{m,n} m \geq 4, n \geq 2$, all vertices are solvable.*

Proof:
Given the proof of Lemma 2.7, we need to consider only the leaf vertices. Let the vertex $v^i_d \in V(P^i_n)$ be at d edges distance from the vertex v_0. We start by considering the first three tentacles P^1_n, P^2_n and P^3_n. Keeping in mind the Theorem 2.3, apply the following procedure:

1. Solve P^1_n in vertex v^1_{n-1}, $(v^2_2 + \cdots + v^2_n)$ in vertex v^2_n and $(v^3_2 + \cdots + v^3_n)$ in vertex v^3_n.
2. Apply $(v^1_{n-1} \xrightarrow{n} v^2_1 \xrightarrow{n+1} v^3_n \xrightarrow{2n} v^2_n)$. Let the remaining vertex $v^* = v^3_1$ be "an anchor".

If $P_i, P_{i+1}, \ldots, P_r, r \geq 2$ tentacles are left, apply the following until only one is left:

1. Solve P^i_n in vertex v^i_{n-1} and $(v^{i+1}_2 + \cdots + v^{i+1}_n)$ in vertex v^{i+1}_n.
2. Apply $(v^i_{n-1} \xrightarrow{n} v^* \xrightarrow{n+1} v^{i+1}_n \xrightarrow{2n} v^2_n)$ and note the new anchor $v^* = v^{i+1}_1$.

When only one P_r tentacle remains, solve it in vertex v^r_{n-1} and apply $(v^r_{n-1} \xrightarrow{n} v^* \xrightarrow{n+1} v^2_n)$. The solution sequence in the leaf vertex v^2_n is now complete. □

Lemma 2.12. *Generalized $S_{(a_m)}, a_1 = \cdots = a_{m-1} = \alpha \geq 2, a_m \geq 2\alpha$ is solvable in the vertex $v^{a_m}_{2\alpha} \in V(P_{a_m})$ at 2α edges distance from the vertex v_0.*

Proof:
We solve paths using the Theorem 2.3. Start by solving the path $(v^{a_m}_{2\alpha} + v^{a_m}_{2\alpha+1} + \ldots)$ in $v^{a_m}_{2\alpha}$. Use the following $(\star)$ procedure to eliminate groups of 6 paths $P^{k_1}_\alpha, \ldots, P^{k_6}_\alpha$ one by one:

1. Solve two paths in $v^{k_{1,2}}_\alpha$, two paths in $v^{k_{3,4}}_{\alpha-1}$, and the last two paths in $v^{k_{5,6}}_\alpha$ minus v^k_1.
2. Apply $(v^{k_3}_{\alpha-1} \xrightarrow{\alpha} v^{k_5}_1 \xrightarrow{\alpha+1} v^{k_6}_\alpha \xrightarrow{2\alpha} v^{k_1}_\alpha \xrightarrow{3\alpha} v^{a_m}_{2\alpha})(v^{k_4}_{\alpha-1} \xrightarrow{\alpha} v^{k_6}_1 \xrightarrow{\alpha+1} v^{k_5}_\alpha \xrightarrow{2\alpha} v^{k_2}_\alpha \xrightarrow{3\alpha} v^{a_m}_{2\alpha})$.

If $6 \mid m-1$, solve $(v_0 + \cdots + v^{a_m}_{2\alpha})$. Otherwise, we are left with $r \in \{1,2,3,4,5\}$ paths. If $r = 1$, then the remaining subgraph is a path. Otherwise, we can reduce it to a path:

- $r \in \{2,3\}$: Solve $(P^{k_1}_\alpha + v_0 + P^{k_2}_\alpha)$ in $v^{k_2}_1$ and apply $(v^{k_2}_1 \xrightarrow{2\alpha+1} v^{a_m}_{2\alpha})$. If $r = 3$, it remains to solve $P^{k_3}_\alpha, P^{a_m}_{1,\ldots,\alpha-1}$ in $v^{k_3}_{\alpha-1}, v^{a_m}_1$ and to apply $(v^{k_3}_{\alpha-1} \xrightarrow{\alpha} v^{a_m}_1 \xrightarrow{2\alpha-1} v^{a_m}_{2\alpha})$.
- $r \in \{4,5\}$: Use $(\star)$ on the first four paths k_1, k_3, k_5, k_6. To eliminate the remaining vertices, use additional vertices $(v^{a_m}_1 + v^{a_m}_2 + \cdots + v^{a_m}_t), t < 2\alpha$ as needed.

□

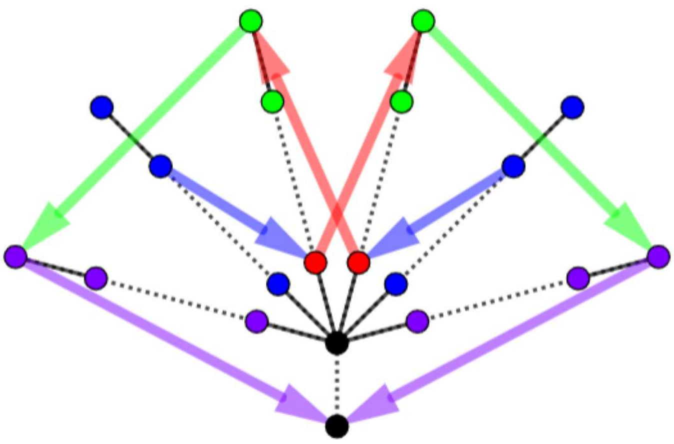

Image 4: Reduction of 6 tentacles by moving all their frogs to the $v_{2\alpha}^{(a_m)}$ vertex.

Theorem 2.13. *Generalized $S_{(a_m)}$ graph is solvable in every vertex if $a_1, a_2, \ldots, a_m > 1$.*

Proof:
We solve paths using the Theorem 2.3. The problem is to solve the graph in an arbitrary vertex $v_d^D \in V(P_n^D)$ at d edges distance from the vertex v_0.

Start by eliminating paths longer than d. That is, without loss of generality, consider only paths $P_{c_1}, \ldots, P_{c_r}$ such that $d \geq c_1 \geq c_2 \geq \cdots \geq c_r$. Find the largest k such that $s_k \sum_{i=1}^{k} c_i \leq d$. Then, the strategy from the proof of the Theorem 2.10 can be used to eliminate groups of paths, also called tentacles. This works as long as $c_k > c_{k+1}$ or $\exists l > 1 : c_k > c_{k+l}$. It also works when $d \neq s_k$, due to the "leaf problem". In other words, two equal tentacles can be combined in any non-leaf vertex, as seen in Lemma 2.7.

Otherwise, $c_k = c_{k+1} = \cdots = c_r = \alpha$ and $d = \alpha + t$, where $t = s_k - c_k$. If $t = 0$, then we have a subgraph $H \simeq S_{R+1,\alpha}$ which is solvable in $v_d = v_\alpha$ due to the Theorem 2.9. Moreover, $t \notin (0, \alpha)$ due to $d \geq c_{k-1} \geq c_k$. It follows that $t \geq \alpha$.

Let $R = r - k + 1 \geq 2$. If $R \in \{2, 3\}$, then we can modify the strategy to use one less vertex v_1^* to avoid the leaf problem. This is not a problem because the remaining subgraph is a path or we can apply $(\{v_0, v_1, \ldots, v_{t-1}\} \xrightarrow{t} v_1^*)(P_\alpha \xrightarrow{\alpha} v_1^* \xrightarrow{\alpha+t+1} v_d)$ to reduce the remaining subgraph to a path. It follows that $R \geq 4$.

If $t \geq 2\alpha$, then $d \geq 3\alpha$ and we can use Lemma 2.11 to eliminate groups of 4 or more tentacles at a time. Otherwise, we can eliminate smaller groups of tentacles except when t is a multiple of α, because of the leaf problem. In other words: If $t > \alpha$, then we can reduce R to $R' \geq 4$ such that $R'\alpha \leq d$.

Now we can modify the strategy to use one less vertex v_1^* to avoid the leaf problem. This is not a problem because we can reduce the remaining subgraph to a path. We do this by solving the subgraph $T \simeq S_{(a_p)}$ in the vertex v_0 and applying $(v_0 \to v_d)$. This is possible because $(R' - 2)\alpha + 1 < d$.

It remains to consider $t = \alpha$. This case was solved by Lemma 2.12. $\square$

It remains to consider the generalized $S_{(a_m)}$ containing indices $i \in I$ such that $a_i = 1$. The subcase $a_1 = \cdots = a_m = 1, a_{m+1} = n$, $S_{(a_{m+1})} \simeq D_{m,n}$ is examined in the next section.

3 Dandelion $D_{m,n}$ graphs

We show that a dandelion $D_{m,n}, m, n \geq 2$ is solvable in every vertex if n is large enough. We have partially solved the other case.

Recall that a dandelion $D_{m,n}$ consists of a star S_m and a path P_n connected at the center vertex of the star. From now on, we use the following notation:

- Let leaf vertices of the star $S_m = S'_m$ subgraph of $D_{m,n}$ be in $V_s = \{s_1, s_2, \dots, s_m\}$ and let v_0 be the center vertex of that subgraph. Let $S'_{(a,b)}$ represent the subgraph induced by $\{s_a, s_{a+1} \dots, s_{b-1}, s_b\} \subseteq V_s$.
- Let vertices of the path $P_n = P'_n$ subgraph of $D_{m,n}$ be in $V_p = \{p_1, p_2, \dots, p_n\}$ such that $p_i, i \in \{1, \dots, n\}$ is at i edges distance from v_0. Let $P'_{(a,b)}$ represent the subgraph induced by $\{p_a, p_{a+1} \dots, p_{b-1}, p_b\} \subseteq V_p$.
- When we use the Theorem 2.3 to solve subgraphs $P'_{(a,b)}$ in p_i, we write $(P'_{(a,b)}(i) \rightarrow \dots)$. When we use the Theorem 2.5 to solve the subgraph $S'_{(a,b)}$ in v_0, we write $(S'_{(a,b)} \rightarrow \dots)$.

Solving the vertex v_0 is trivial.

Lemma 3.1. *In a dandelion $D_{m,n}$, the vertex of degree $m + 1$ is solvable.*

Proof:
According to the Theorem 2.5 we can solve the star S_m subgraph in its central vertex v_0. According to Lemma 2.2 we can solve the path P_n subgraph in its leaf vertex v_0. Therefore, a dandelion $D_{m,n}$ is solvable in the vertex v_0 of degree $m + 1$. □

It remains to solve the vertices in S'_m and P'_n. Looking at the vertices in the path subgraph, it is not hard to see that we only need to consider the $P'_{(1,m)}$ subgraph.

Theorem 3.2. *In a dandelion $D_{m,n}, m, n \geq 2$, vertices $p_i, i \in \{m + 1, m + 2, \dots, n\}$ of the path P'_m subgraph are solvable. If $m = 2$, then p_m is also solvable.*

Proof:
For $t \in \mathbb{Z}^*$, solution sequences can be constructed as follows:

1. Use Lemma 3.1 to solve subgraph $(S'_{(1,m)} + P'_{(1,t)}) \simeq D_{m,t}$ in vertex v_0.
2. Use the Theorem 2.3 to solve subgraph $P'_{(t+1,n)}$ in p_{m+t+1}.
3. Apply the final transition $(v_0 \xrightarrow{m+t+1} p_{m+t+1})$.

Specially for $m = 2$, we can have $t = -1$ by solving the first step in s_1 instead of in v_0 because subgraph $S'_{(1,2)} \simeq S_2 \simeq P_3$ is a path to which Theorem 2.3 applies. □

In the following subsection, we solve the leaf vertices of S'_m subgraph for $n \geq m$. Otherwise, it is not clear how to construct the solution sequences.

In the other subsection, we solve the vertices of $P'_{(1,m)}$ subgraph for $n \geq 2m+3$. Otherwise, it is difficult to find solvable vertices.

3.1 Solving vertices of Star $S'_m \subset D_{m,n}$ subgraph

In this subsection, we consider leaf vertices. First, we handle the case $n \geq m$. It is not hard to exhaust reachable problem states to verify that three cases (m, n) are not solvable.

Lemma 3.3. *In a dandelion $D_{m,n}, m \geq 2, n \geq m$, leaf vertices of the star S'_m subgraph are not solvable if $(m, n) \in \{(3,4), (3,5), (4,6)\}$.*

Proof:
By exhausting all reachable problem states. □

We show that all other cases $(n, m), n \geq m$ are solvable. Note that $S'_{(1,2)} \simeq S_2 \simeq P_3$. As a result, it is easy to resolve $m = 2$.

Lemma 3.4. *In a dandelion $D_{2,n}, n \geq 2$, leaf vertices of the star S'_2 subgraph are solvable.*

Proof:
Solution sequences can be constructed as follows:

1. Use the Theorem 2.3 to solve subgraph $P'_{(1,n)}$ in vertex p_{n-1}.
2. Apply transitions $(v_0 \xrightarrow{1} s_1 \xrightarrow{2} s_2)$ and $(p_{n-1} \xrightarrow{n} s_2)$.

□

Since $n \geq m$, we can write $n = m + t, t \in \mathbb{Z}^*$. The following lemma will resolve cases $t \in \{0, 1, 2\}$ not covered by previous two lemmas.

Lemma 3.5. *In a dandelion $D_{m,n}, m \geq 2, n \geq m$, leaf vertices of the star S'_m subgraph are solvable if $n = m + t, m \geq t + 3, t \in \mathbb{Z}^*$.*

Proof:
Solution sequences can be constructed as follows:

1. Use Lemma 3.1 to solve subgraph $(S'_{(1,m-1)} + P'_{(1,t)}) \simeq D_{m-1,t}$ in vertex v_0.
2. Use the Theorem 2.3 to solve subgraph $P'_{(t+1,n-1)} \simeq P_{m-1}$ in vertex p_{m-2}.
3. Apply transitions $(p_{m-2} \xrightarrow{n-1-t} s_m)$ and $(v_0 \xrightarrow{m+t} p_n \xrightarrow{n+1} s_m)$.

□

Note that the previous lemma does not consider the cases $m < t + 3, t \in \mathbb{Z}^*$. For $t \in \{0, 1, 2\}$ these cases are not solvable unless $m = 2$, as mentioned in the previous lemmas. It remains to consider $n \geq m + 3$ for $m \geq 3$.

To solve the remaining case, the following lemma considers two strategies depending on whether n is greater than $2m$ or not. Both are similar to the previous lemma, but use different groupings of vertices.

Lemma 3.6. *In a dandelion $D_{m,n}, m \geq 3, n \geq m+3$, leaf vertices of S'_m are solvable.*

Proof:
For $n > 2m$, solution sequences can be constructed as follows:

1. Use Lemma 3.1 to solve subgraph $S'_{(1,m-1)}$ in vertex v_0.
2. Use the Theorem 2.3 to solve $P'_{(1,m-1)}$ in p_{m-2}, $P'_{(m,n-m-1)}$ in p_m and $P'_{(n-m,n)}$ in p_n.
3. Apply transitions $(v_0 \xrightarrow{m} p_m \xrightarrow{n-m} p_n \xrightarrow{n+1} s_m)$ and $(p_{m-2} \xrightarrow{m-1} s_m)$.

Otherwise, for $n \leq 2m$, solution sequences can be constructed as follows:

1. Use Lemma 3.1 to solve subgraph $S'_{(1,m-1)}$ in vertex v_0.
2. Use the Theorem 2.3 to solve $P'_{(1,m-1)}$ in p_{n-m-1} and $P'_{(m+1,n)}$ in p_{n-2}.
3. Apply transitions $(v_0 \xrightarrow{m} p_m \xrightarrow{m+1} s_m)$ and $(p_{n-m-1} \xrightarrow{m-1} p_{n-2} \xrightarrow{n-1} s_m)$.

□

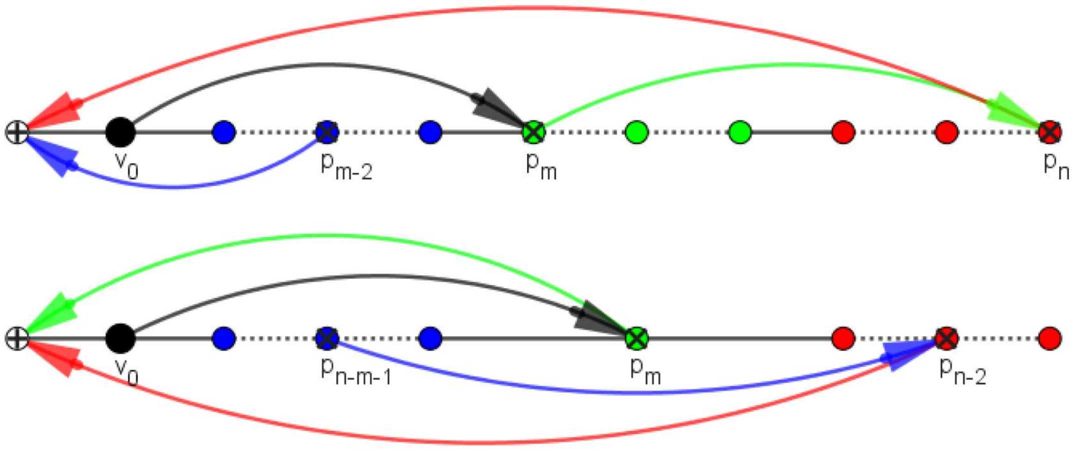

Image 5: Strategies used in Lemma 3.6.

Theorem 3.7. *In a dandelion $D_{m,n}, n \geq m$, all leaf vertices of the star S'_m subgraph are solvable if and only if $(m,n) \notin \{(3,4),(3,5),(4,6)\}$.*

Theorem 3.7 is a consequence of Lemmas 3.3, 3.5, 3.4, 3.6. It remains to consider the case $n < m$. This is hard because proving that a solution sequence does not exist is hard. We state the following problem for $n = m-k, k \in \{1,2,\ldots,m-2\}$.

Problem 3.8 (The dandelion leaf problem.)**.** *Let $M_k, k \in \mathbb{N}$ be the set of all m such that vertex $s \in V_s$ of dandelion $D_{m,m-k}$ is solvable. Can we find M_k for all k?*

The first step in solving the problem would be to find $m_k = \min M_k$. It is not hard to observe an upper bound.

Proposition 3.9. *It holds that* $m_1 \leq 14$ *and* $m_{k+1} \leq 3m_k - 2k + 5$, $\forall k \in \mathbb{N}$.

Proof:
If $k = 1$, it is not hard to solve $D_{14,13}$ in vertex $s \in V_s$. WLOG $s = s_{13}$, then we can:

1. Apply transitions $(S'_{(1,12)} \xrightarrow{12} v_0 \xrightarrow{13} p_{13} \xrightarrow{14} s_{13})$ and solve $P'_{(5,7)}(6)$ and $P'_{(8,12)}(12)$.

2. Apply transitions $(p_6 \xrightarrow{3} p_3 \xrightarrow{4} s_{14} \xrightarrow{5} p_4)(p_1 \xrightarrow{1} p_2 \xrightarrow{2} p_4)$ then $(p_4 \xrightarrow{8} p_{12} \xrightarrow{13} s_{13})$.

We can now inductively construct an upper bound for $m_{k+1}, k \in \mathbb{N}$. Take solution sequence of D_{m_k,m_k-k} solved in $s \in V_s$. Remove the transitions $(S'_{(1,m_k-k-1)} \to v_0 \to p_{m_k-k} \to s)$ and the transition $(s_i \to v_0), s_i \neq s$ for one $s_i \in V_s$. Now define $D'_{m,m-k} = D_{m_{k+1},3(m_k-k)+4}$,

$$m_{k+1} - (k+1) = 3(m_k - k) + 4 \implies m_{k+1} = 3m_k - 2k + 5.$$

We can apply the transitions $(S'_{(1,m_{k+1}-(k+1)-1)} \to v_0 \to p_{3(m_k-k)+4} \to s)$ in place of the removed transitions. It remains to take care of the vertices that were added. Namely, we need to move frogs from $p_{m_k-k}, \ldots, p_{3(m_k-k)+3}$ to s and a frog from s_i to s. This can be done as follows:

1. Apply transitions $(P'_{(m_k-k+2,2(m_k-k)+1)}(2(m_k-k)) \xrightarrow{m_k-k} p_{m_k-k} \xrightarrow{m_k-k+1} s_i)$.

2. Apply transitions $(s_i \xrightarrow{m_k-k+2} p_{m_k-k+1} \xrightarrow{m_k-k+3} p_{2(m_k-k)+4})$.

3. Apply transitions $(P'_{(2(m_k-k)+2,3(m_k-k)+3)}(2(m_k-k)+4) \xrightarrow{2(m_k-k)+5} s)$.

We now have a solution sequence for $D'_{m,m-k} = D_{m_{k+1},m_{k+1}-(k-1)}$. □

The previous proposition is just a simple upper bound. That is, the exact value of m_k could be much smaller. For example, the proposition gives $m_2 \leq 45$, but we found $m_2 \leq 32$ using a different strategy. We have not been able to find a general strategy for the best possible upper bound. Therefore, Problem 3.8 remains open.

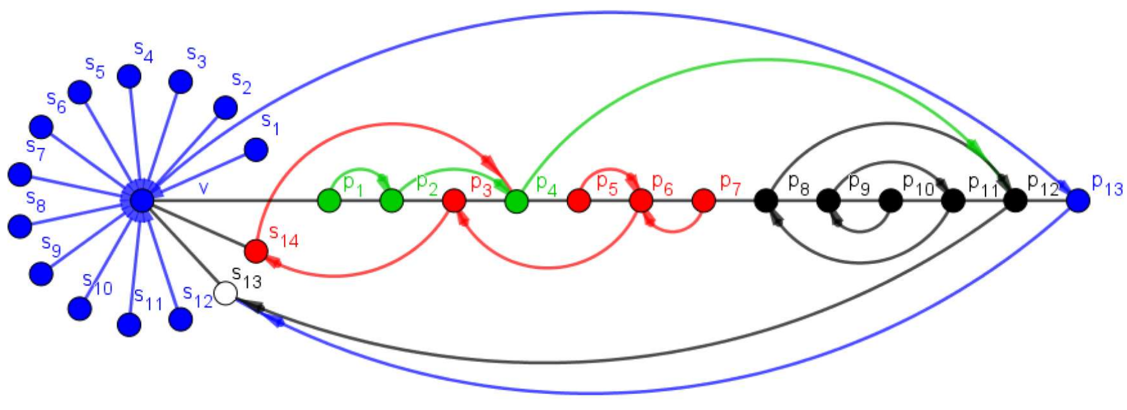

Image 6: Solution sequence for s_{13} vertex of dandelion graph $D_{14,13}$.

3.2 Solving vertices of Path $P'_n \subset D_{m,n}$ subgraph

According to the Theorem 3.2 we need to consider only the $P'_{(1,m)}$ subgraph. Since for $D_{2,n}$ we only need to consider whether the vertex p_1 is solvable or not, we can first solve this case as a separate theorem.

Theorem 3.10. *In a 2-dandelion $D_{2,n}, n \geq 2$, all vertices are solvable if and only if $n \geq 6$. Otherwise, for $n \in \{2,3,4,5\}$, all vertices except p_1 are solvable.*

Proof:
The fact that vertices $V \setminus \{p_1\}$ are always solvable follows from Theorems 3.2, 3.7. The fact that p_1 is not solvable for $n < 6$ can be verified by exhausting all sequences of reachable problem states.

For $n \geq 11$ and $t = n - 11$, solution sequences for p_1 can be constructed as follows:

1. Use Lemma 2.2 to solve $P'_{(6,6+t)}$ in p_6 and $P'_{(7+t,n)}$ in p_n.
2. Apply transitions $(v_0 \xrightarrow{1} s_1 \xrightarrow{2} s_2 \xrightarrow{3} p_2 \xrightarrow{4} p_6 \xrightarrow{5+t} p_n \xrightarrow{n-1} p_1)$ and $(\{p_3, p_5\} \xrightarrow{2} p_4 \xrightarrow{3} p_1)$.

It is left to consider $n \in \{6,7,8,9,10\}$. We give the following solution sequences:

$$
\begin{aligned}
G^{f_x^6} &= G^{f_0}(S'_2(s_2) \xrightarrow{3} p_2 \xrightarrow{4} p_6 \xrightarrow{5} p_1)(P'_{(3,5)}(4) \xrightarrow{3} p_1),\\
G^{f_x^7} &= G^{f_0}(p_3 \xrightarrow{1} p_2)(S'_2(s_2) \xrightarrow{3} p_2 \xrightarrow{5} p_7 \xrightarrow{6} p_1)(P'_{(4,6)}(4) \xrightarrow{3} p_1),\\
G^{f_x^8} &= G^{f_0}(p_2 \xrightarrow{1} p_3)(S'_2(v_0) \xrightarrow{3} p_3 \xrightarrow{5} p_8)(p_7 \xrightarrow{1} p_8 \xrightarrow{7} p_1)(P'_{(4,6)}(4) \xrightarrow{3} p_1),\\
G^{f_x^9} &= G^{f_0}(S'_2(s_2) \xrightarrow{3} p_2 \xrightarrow{4} p_6 \xrightarrow{5} p_1)(P'_{(3,5)}(4))(P'_{(7,9)}(7))(p_4 \xrightarrow{3} p_7 \xrightarrow{6} p_1),\\
G^{f_x^{10}} &= G^{f_0}(S'_2(s_2) \xrightarrow{3} p_2 \xrightarrow{4} p_6 \xrightarrow{5} p_1)(P'_{(3,5)}(5))(P'_{(7,10)}(8))(p_5 \xrightarrow{3} p_8 \xrightarrow{7} p_1),
\end{aligned}
$$

by applying Theorem 2.3 to solve $P'_{(a,b)}$ in p_i as $P'_{(a,b)}(i)$ and $S'_2 \simeq P_3$ in s_2, v_0 as $S'_2(s_2), S'_2(v_0)$. $\square$

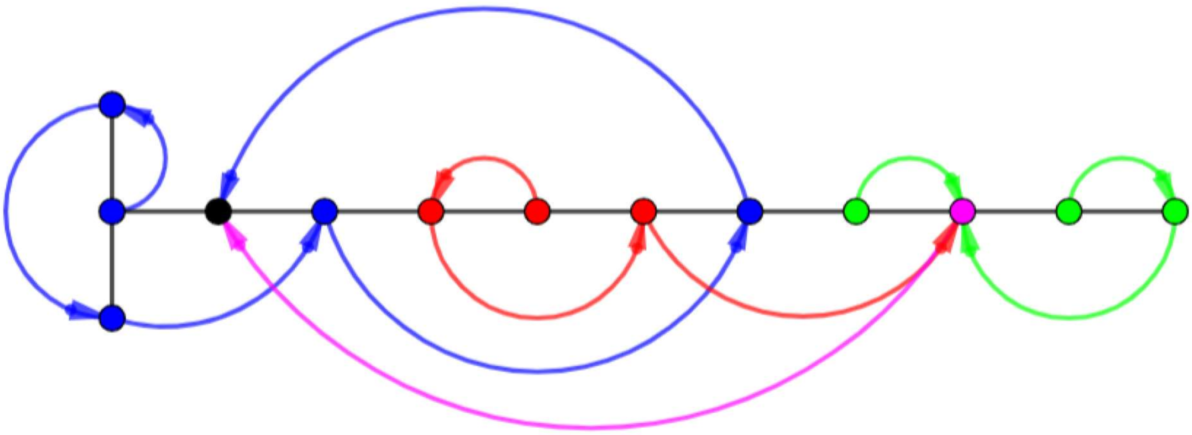

Image 7: Solution sequence for p_1 vertex of dandelion graph $D_{2,10}$.

It is left to consider vertices of $P'_{(1,m)}$ subgraph for $m \geq 3$. First, we construct solution sequences for $n = 2m+3$. In that case, solving the vertex $p_{\frac{m}{2}}$ for even m must be considered separately. For example, the following image illustrates the case $m = 6$.

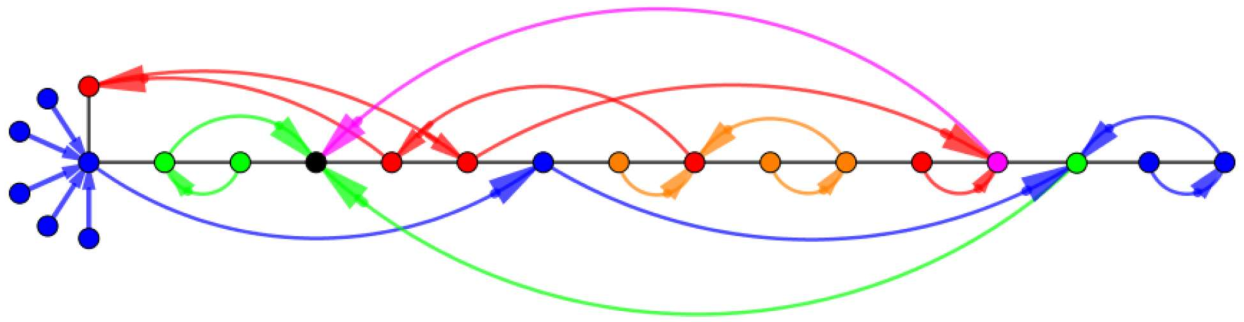

Image 8: Solution sequence for p_3 vertex of dandelion graph $D_{6,15}$ (see [6]).

A similar strategy can be used for even $m > 6$. This strategy was inspired by the construction given in the Proposition 3.9.

Lemma 3.11. *In a dandelion $D_{m,n}, m$ even, vertex $p_{\frac{m}{2}}$ is solvable if $n = 2m+3, m \geq 6$.*

Proof:
For $m = 6$, see Image 8. For $m \geq 8$, let $S''_m(v_0) = S'_{(1,m-1)}(v_0) = (\{s_1, \ldots, s_{m-1}\} \xrightarrow{m-1} v_0)$ and follow the procedure:

1. $G^{h_1} = G^{f_0}(S''_m(v_0) \xrightarrow{m} p_m \xrightarrow{m+1} p_{2m+1})(P'_{(2m-\frac{m}{2}+2,2m+1)}(2m+1) \xrightarrow{m+\frac{m}{2}+1} p_{\frac{m}{2}})$.
2. $G^{h_2} = G^{h_1}(P'_{(m+1,2m-\frac{m}{2}+1)}(m+2) \xrightarrow{\frac{m}{2}+1} p_{\frac{m}{2}+1} \xrightarrow{\frac{m}{2}+2} s_m \xrightarrow{\frac{m}{2}+3} p_{\frac{m}{2}+2})(p_n \xrightarrow{1} p_{n-1})$.
3. $G^{f_x^{p_{m/2}}} = G^{h_2}(P'_{(1,\frac{m}{2}-1)}(3) \xrightarrow{\frac{m}{2}-1} p_{\frac{m}{2}+2})(P'_{(\frac{m}{2}+2,m-1)}(\frac{m}{2}+2) \xrightarrow{2m-\frac{m}{2}} p_{2m+2} \xrightarrow{2m-\frac{m}{2}+2} p_{\frac{m}{2}})$.

$\square$

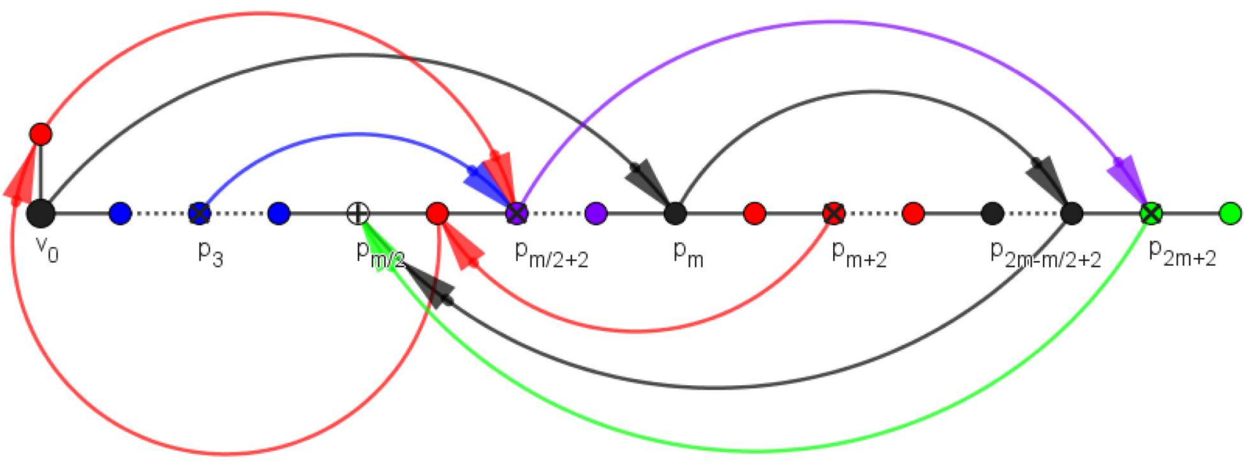

Image 9: Illustration of the strategy used in Lemma 3.11.

Lemma 3.12. *In a dandelion $D_{m,n}, m \geq 3$, all vertices are solvable if $n = 2m+3$, except vertex p_2 if m is exactly equal to $m = 4$.*

Proof:
We only need to consider $v \in V(P'_{(1,m)})$ due to Lemma 3.1 and Theorems 3.2, 3.7. For $t \in [0, m)$, let

$$G^{f_1} = G^{f_0}(S'_m \xrightarrow{m+1} p_{m+1} \xrightarrow{m+2} p_n)(P'_{(n-t,n)}(n) \xrightarrow{m+3+t} p_{m-t}).$$

Let $G^{f_x} = G^{f_x^{p_{m-t}}}$ and $r = n - 2(t+1)$. Now we can solve vertices $p_{m-t}, t \in [0, \frac{m}{2})$ as

$$G^{f_x} = G^{f_1}(P'_{(m+2,n-t-1)}(r) \xrightarrow{m+1-t} p_{m-t})(P_{1,m}(m-t)).$$

We can solve remaining vertices $p_{m-t}, t \in [p_{\lfloor \frac{m}{2} \rfloor}, m)$ as

$$G^{f_x} = G^{f_1}(P'_{(m-t+1,m)}(r) \to p_{n-t-2})(P'_{(m+2,n-t-1)}(n-t-2) \to p_{m-t})(P'_{(1,m-t)}(m-t)),$$

unless m is even and $t = \frac{m}{2}$. In that case, $r = n - 2(t+1) = m+1$ but $f_1(p_{m+1}) = 0$ as seen in G^{f_1}. Therefore, It remains to solve the $p_{\frac{m}{2}}$ vertex for even m as a special case. It is not hard to verify that this is not possible for $m = 4$. Otherwise, see Lemma 3.11. $\square$

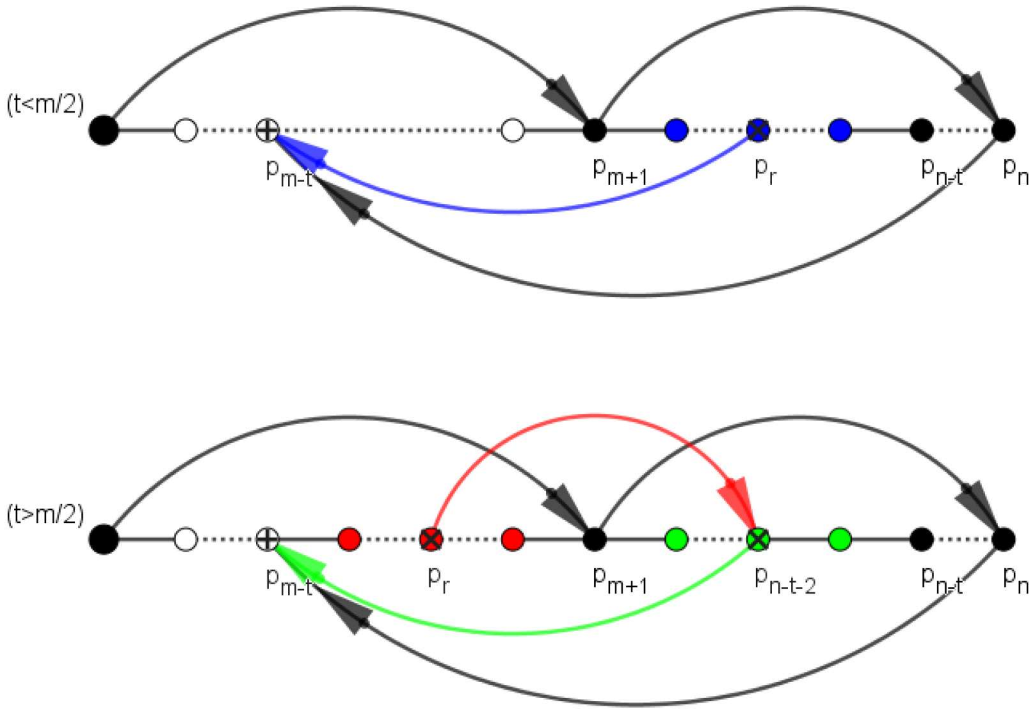

Image 10: Strategies used in Lemma 3.12.

Note that in the proof of Lemma 3.12, p_r comes either after or before p_{m+1}. In the case where $p_r = p_{m+1}$, we must use Lemma 3.11 instead because we cannot use the same vertex twice. For this reason, $p_{\frac{m}{2}}$ had to be considered separately.

Corollary 3.13. *In a dandelion $D_{m,n}, m \geq 3,$ all vertices are solvable if $n \geq 4m+6$.*

Proof:
We only need to consider $p_i \in V(P'_{(1,m)})$ due to Lemma 3.1 and Theorems 3.2, 3.7.

Dandelion graph $G = D_{m,n}, m \geq 3$ for $n = 4m+6+t, t \in \mathbb{Z}^*$ can be split into two disjoint subgraphs $G_1 \simeq D_{m,2m+3}$ and $G_2 = P'_{(2m+4,4m+6+t)} \simeq P_{2m+3+t}$. We know that G_1 is solvable in any vertex according to Lemma 3.12 and that G_2 is solvable in any vertex according to the Theorem 2.3. Solve G in vertex $p_i \in V(P'_{(1,m)}) \subset V(G_1)$ as follows:

1. Solve G_1 in p_i and G_2 in $p_{2m+3+t+i}$.
2. Apply transition $(p_{2m+3+t+i} \xrightarrow{2m+3+t} p_i)$.

□

It remains to consider the case $n \in (2m+3, 4m+6)$ except when $(m,n) = (3,10)$. It also remains to see if it is possible for some vertices to be solvable when $n < 2m+3$.

Theorem 3.14. *In a dandelion $D_{m,n}, m \geq 3,$ all vertices are solvable if $n \geq 2m+3$, except vertex p_2 if $(m,n) = (4,11)$ and vertex p_1 if $(m,n) = (3,10)$.*

Proof:
Due to Corollary 3.13, we only need to consider $n \in (2m+3, 4m+6)$ and $p_i \in V(P'_{(1,m)})$. Using similar argument as in proof of Corollary 3.13, we only need to consider solving vertices $p_{m-t}, t \in [0,m)$ for $n \in (2m+3, 3m+7+t) = N_0$. Let $n = 2m+3+k, k \in \mathbb{N}$ and $y = n-t$. Start with the following problem state:

$$G^{f_1} = G^{f_0}(S'_m(v_0) \to p_{m+1})(P'_{(1,m-t)}(m-t)).$$

First we solve the case $k < t$, which implies $t \geq 2$. If $m = 3$, it is not hard to verify that $(m,n) = (3,10)$ is not solvable. If $t = 2$, then $k = 1$ and use the following solution sequence:

$$G^{f_x^{p_{m-2}}} = G^{f_1}(P'_{(m-2,m)}(m-2))(p_{m+1} \to P'_{(n-2,n)}(n-1) \to p_{m-2})(P'_{(m+2,n-3)}(2m-2) \to p_{m-2}).$$

Otherwise, for $m \geq 4, k \in [1,t), t \in [3,m)$, let $\delta = m+k-2t$. Then, solution sequence is:

1. If $\delta > 0$, then $G^{h_1} = G^{f_1}(p_{m+1} \to P'_{(n-t,n)}(2m+3) \to p_{m-t})$ and
$$G^{f_x^{p_{m-t}}} = G^{h_1}(P'_{(m-t,m)}(m-t))(P'_{(m+2,n-t-1)}(m+1+\delta) \to p_{m-t}).$$
2. If $\delta = 0$, then $G^{h_2} = G^{f_1}(p_{m+2} \to p_{m+1} \to P'_{(n-t,n)}(2m+4) \to p_{m-t})$ and
$$G^{f_x^{p_{m-t}}} = G^{h_2}(P'_{(m-t+1,m)}(m) \to P'_{(m+3,n-t-1)}(m+t) \to p_{m-t}).$$
3. If $\delta < 0$, then $G^{h_3} = G^{f_1}(p_{m+1} \to P'_{(n-t,n)}(2m+3) \to p_{m-t})$ and
$$G^{f_x^{p_{m-t}}} = G^{h_3}(P'_{(m-t+1,m)}(m+1+\delta) \to P'_{(m+2,n-t-1)}(n-t-2) \to p_{m-t}).$$

It is not hard to verify that these solution sequences always work for the case $k < t$.

Otherwise, $k \geq t$. Let $k = t + k_0 + k_1 + k_2$ where $k_0, k_1, k_2 \in \mathbb{Z}^*$. We consider three cases:

1. Set $k_1 = k_2 = 0$ and allow $k_0 \in [0, t]$. That is, we have $n \in [2m + 3 + t, 2m + 3 + 2t]$. In this case, we first move $t - k_0$ frogs to p_{m-t}, then we move k_0 frogs to p_{m+1} to be able to bring the farthest $t + 1$ frogs from vertices $\{p_y, p_{y+1} \dots, p_n\}$ to p_{m-t}.

$$\begin{aligned} G^{f_2^{(1.)}} &= G^{f_1}(P'_{(m-t,m-k_0)}(m-t))(P'_{(m+1-k_0,m+1)}(m+1))(P'_{(y,n)}(y)), \\ G^{f_3^{(1.)}} &= G^{f_2^{(1.)}}(p_{m+1} \to p_y \to p_{m-t}). \end{aligned}$$

Now only $G^{f_x^{p_{m-t}}} = G^{f_3^{(1.)}}(P'_{(m+2,y-1)}(2m - t + 1 + k_0) \to p_{m-t})$ remains, which moves the remaining $m + 1 + k - t$ frogs to p_{m-t}.

2. Set $k_2 = 0, k_0 = t$ and allow $k_q \in [1, m)$. That is, $n \in [2m + 3 + 2t + 1, 3m + 2 + 2t]$. In this case, in addition to $k_0 = t$ frogs that we move as in 1., we also need to move additional k_1 frogs to p_{m+1} to be able to bring the farthest $t + 1$ frogs to p_{m-t}.

$$G^{f_2^{(2.)}} = G^{f_2^{(1.)}}(P'_{(m+2,m+1+k_1)}(m+1))(p_{m+1} \to p_y \to p_{m-t}).$$

Now only $G^{f_x^{p_{m-t}}} = G^{f_2^{(2.)}}(P'_{(m+2+k_1,y-1)}(2m+1) \to p_{m-t})$ remains.

3. Set $k_1 = m - 1, k_0 = t$ and allow $k_2 \in \mathbb{N} \implies k \geq 2t + m$. That is, $n \geq 3m + 3 + 2t$. Although this case holds for any t, it is only necessary if $t \in \{0, 1, 2, 3\}$, because otherwise, $n \notin N_0$. Let $k_2 - 1 = k_2^* \in \mathbb{Z}^*$. We want to move $r = k_0 + k_1 + k_2 = t + m + k_2^*$ frogs from vertices $\{p_{y-1}, p_{y-2}, \dots, p_{y-r}\}$ to p_{m+1}, to be able to bring the farthest $t + 1$ frogs to p_{m-t}. Note that $p_{y-r} = p_{2m+3}$ and that p_{m+1} is always reachable if $t \geq 2$,

$$\begin{aligned} G^{f_2^{(3.a)}} &= G^{f_1}(P'_{(y-r,y-1)}(2m + 1 + t + k_2^*) \to p_{m+1} \to p_y)(P'_{(y,n)}(y) \to p_{m-t}), \\ G^{f_x^{p_{m-t}}} &= G^{f_2^{(3.a)}}(P'_{(m-t,m)}(m-t))(P'_{(m+2,2m+2)}(2m + 1 - t) \to p_{m-t}). \end{aligned} \tag{3.a}$$

Otherwise, if $t = 1$, then it is reachable if we include one more frog,

$$\begin{aligned} G^{f_2^{(3.b)}} &= G^{f_1}(P'_{(y-r-1,y-1)}(2m + 3 + k_2^*) \to p_{m+1} \to p_n)(P'_{(y,n)}(n) \to p_{m-t}), \\ G^{f_x^{p_{m-t}}} &= G^{f_2^{(3.b)}}(P'_{(m-t,m)}(m-t))(P'_{(m+2,2m+1)}(2m-1) \to p_{m-t}). \end{aligned} \tag{3.b}$$

We cannot use a similar trick for $t = 0$ right away, because in this case, $y = n$. This means that including more frogs to compensate for small n would make the transition $(p_{m+1} \to p_i)$ jump to $i > n$. Instead, for $k_2^* \in \{0, 1, 2\}$ we can solve it as

$$G^{f_x^{p_{m-t}}} = G^{f_1}(p_{m+1} \to p_{2m+3} \to p_m)(P'_{(m+2,2m+2)}(2m + k_0^*) \to P'_{(2m+4,n)}(n-2) \to p_m).$$

It remains to solve $t = 0$ for $k_2^* \geq 3$ which is $n \geq 3m + 6$. Actually, we are done with this case, because this n is now large enough for the $(3.a)$ strategy. Even so, note that such a strategy is only needed for $k_2^* = 3$, because $n \notin N_0$ if $k_2^* \geq 4, t = 0$.

This completes the proof. □

We can visualize the previous theorem with the following illustrations.

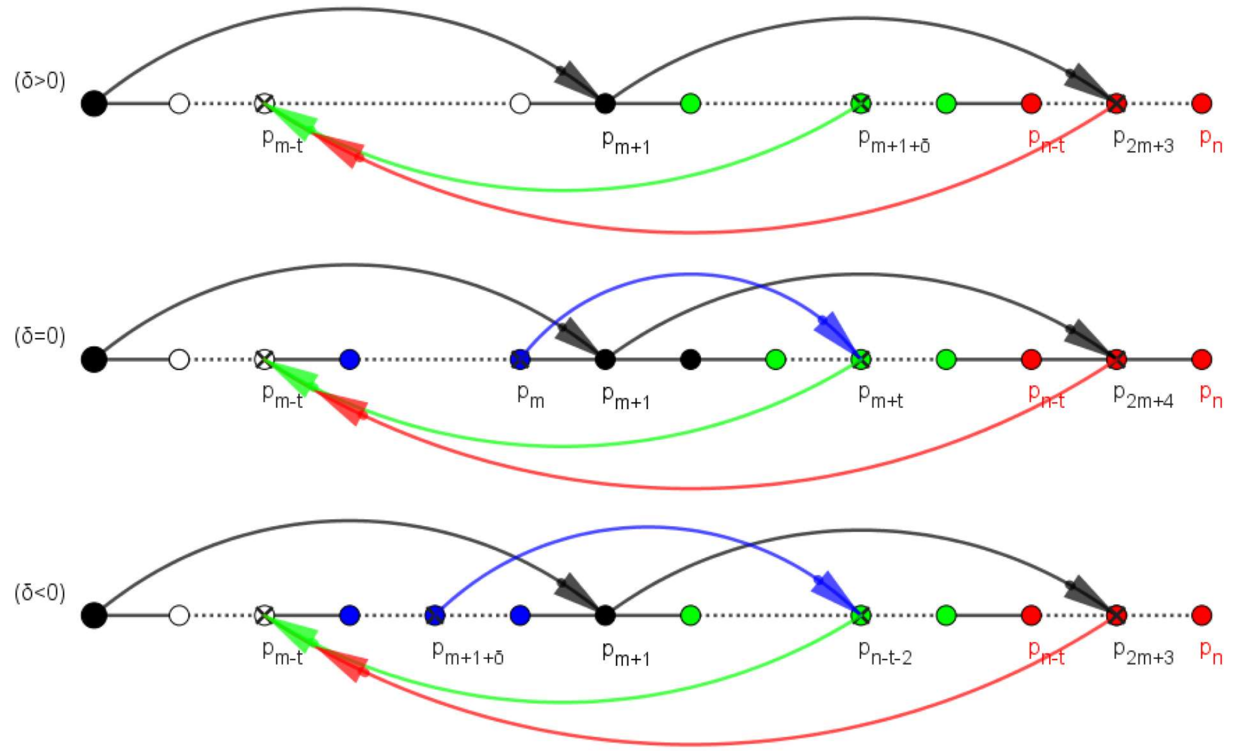

Image 11: Strategies used in Theorem 3.14 for the case $k < t$.

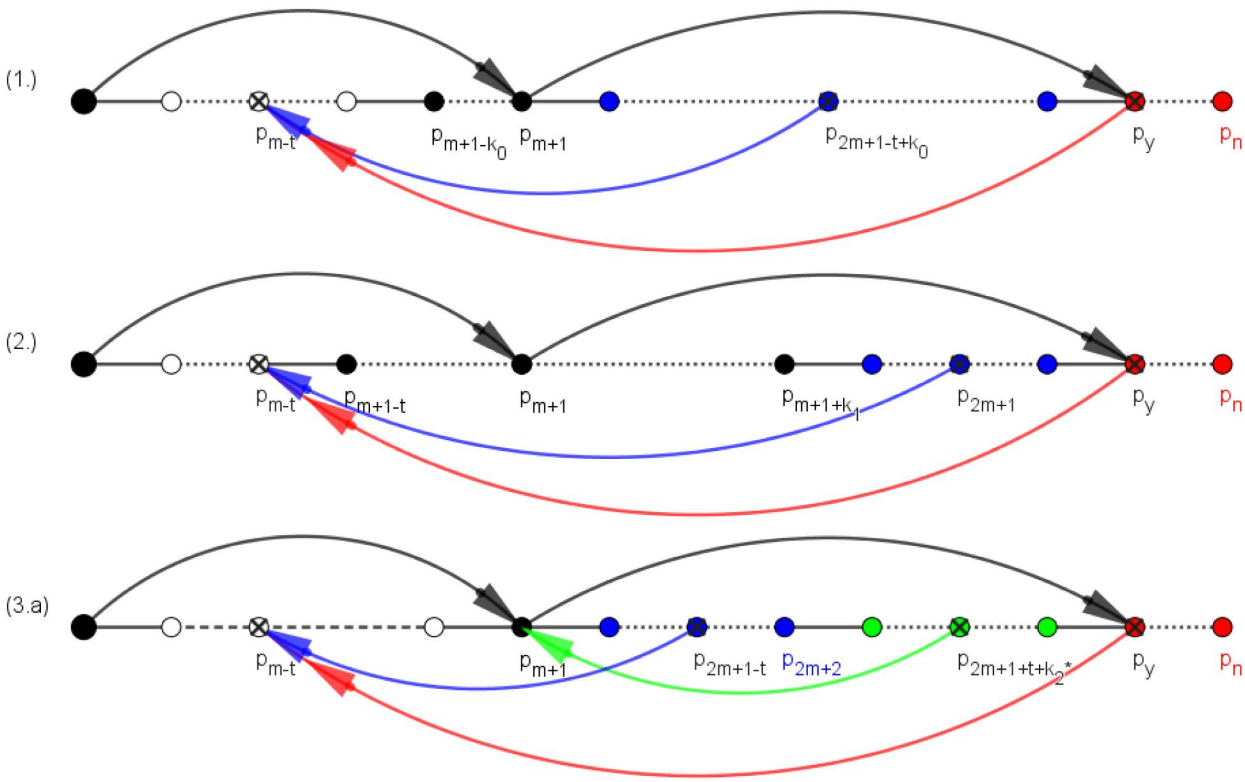

Image 12: Strategies used in Theorem 3.14 for the case $k \geq t$.

It remains to consider vertices $p_i \in V(P'_{(1,m)})$ and $n \leq 2m+2, m \geq 3$. Similarly as in Problem 3.8, the hard part is characterizing unsolvable vertices.

Problem 3.15. *In dandelion $D_{m,n}, n \leq 2m+2$, which $p_i \in V(P'_{(1,m)})$ are solvable?*

For example, we solved $m \in \{3,4,5,6,7\}$ using an exhaustive computer search. Therefore, Problem 3.15 remains open for $m \geq 8$. To summarize:

- $(m=3)$: There are no solvable vertices.
- $(m=4)$: If $n \in \{7,8\}$ then p_4 is solvable.
- $(m=5)$: If $n \in \{9,10,11\}$ then p_5 is solvable.
- $(m=6)$: Solvable cases are $(n \geq 10, \{p_6\}), (n=13, \{p_1\}), (n=14, \{p_1,p_2\})$.
- $(m=7)$: Solvable cases are $(n \geq 10, \{p_7\}), (n=15, \{p_6\}), (n=16, \{p_1,p_2\})$.

4 Unsolvable tree graphs

Graphs that are unsolvable in every vertex can be hard to find. But, there is a trivial class of tree graphs that we can characterize. Let $S_{\{m_1,m_2\}}$ be a "2-star" graph, consisting of two star graphs S_{m_1}, S_{m_2} with centers v_0^1, v_0^2 connected by an edge $\{v_0^1, v_0^2\}$.

Theorem 4.1. *All vertices of a 2-star $S_{\{m_1,m_2\}}, m_1, m_2 \geq 2$ are unsolvable, except when $\{m_1,m_2\} \in \{\{2,2\},\{2,3\}\}$.*

Proof:
Due to symmetry, we need to consider only the $(s_i^1 + v_0^1 + v_0^2 + s_j^2)$ subgraph for any pair of vertices $s_i^1 \in V(S_{m_1}), i = 1,2,\ldots,m_1$ and $s_j^2 \in V(S_{m_2}), j = 1,2,\ldots,m_2$.

It is not solvable in a center $v_0 \in \{v_0^1, v_0^2\}$ because the leaves connected to the opposite center are at a distance of 2 but share their only neighbor.

A solution sequence that solves a leaf vertex cannot combine the centers. Otherwise, the remaining leaves are left without nonzero neighbors. Thus, only transitions beginning with $(v_0^1 \to s_i^1)$, $(v_0^2 \to s_i^2)$, $(s_i^1 \to v_0^1)$, or $(s_i^2 \to v_0^2)$ remain. Considering all such transitions, we see that this only works for $\{m_1,m_2\} \in \{\{2,2\},\{2,3\}\}$. $\square$

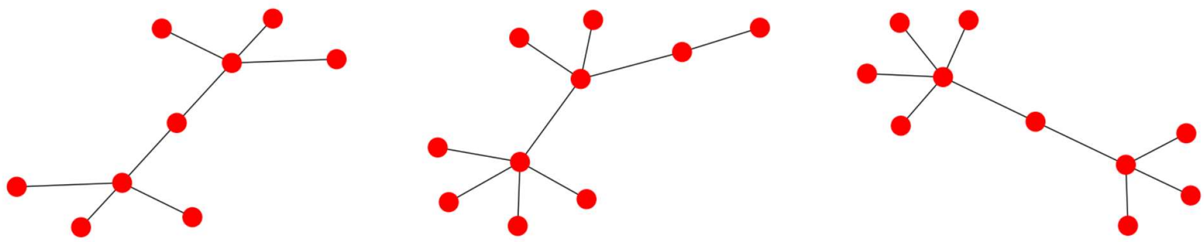

Image 13: All 3 fully unsolvable trees with $|V| \leq 10$ and not covered by the Theorem 4.1.

5 Complete binary tree graphs

Let T_h be a complete binary tree with vertices in V and height of h. Then $|V| = 2^{h+1} - 1$. Let v_0 be the root vertex and $v(i,j) = v_{i,j} \in V$ be the ith vertex at distance j from v_0, $j \in \{1, 2, 3, \dots, h\}$, $i \in \{1, 2, 3, \dots, 2^j\}$. Due to symmetry, we only need to consider the root vertex and one vertex from each V_j set, where $V_j = \{v_{i,j} : i \in \{1, 2, 3, \dots, 2^j\}\}$. For example,

1. $T_1 \simeq P_3$ is solvable in every vertex.
2. T_2 is solvable only in $v_1 \in V_1$.
3. T_3 is solvable in every vertex except in $v_1 \in V_1$.

The above cases can be verified by exhausting all reachable states. Otherwise, we have $h \geq 4$ and the following conjecture.

Conjecture 5.1. *Every vertex of every $T_h, h \geq 4$ is solvable.*

We strongly believe in the conjecture, but are unable to complete the proof. We discovered strategies to solve $T_h, h \leq 20$ in the root vertex v_0 by hand, but we were not able to extend them to a general strategy (see [9, 10]). Similar strategies can be used to solve other vertices, but this is still not a general strategy for $h \geq 4$.

For example, we can reduce the process of solving T_h in a vertex v to solving a smaller case T_{h-k} in vertex v if we can "shave off" top layers of vertices by sending their frogs to vertex v without using frogs in other layers.

We have thought about applying induction to subgraphs, but have not been able to find sufficient strategies or patterns to complete the argument. The main problem with induction is that the number of vertices grows exponentially, but the diameter grows linearly. Consequently, we need to consider small subgraphs that are not necessarily complete binary trees. Maybe someone else is interested in this paper and can follow up on this conjecture.

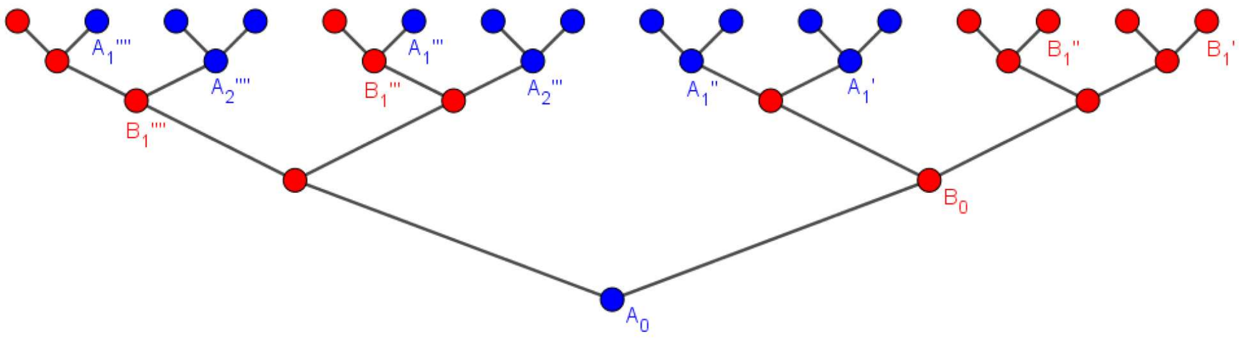

Image 14: Sending frogs from T_4 subtrees to vertex v_0 to "shave off" top 5 layers of T_{19}.

For example, in Image 14 we consider T_4 subtrees from the top 5 layers of a T_{19} tree and select vertices $A_0 \in V_{j=15}$ and $B_0 \in V_{j=16}$ that are 15 and 16 edges away from vertex v_0. Note that $15 + 16 = 31 = |V(T_4)|$. That is, we can then use the transitions $(\dots \to A_0 \to v_0)$ and $(\dots \to B_0 \to v_0)$ to remove all frogs from the top T_4 subtrees. If we were to build an inductive argument, this shows that T_{19} is solvable in v_0 if T_{14} is solvable in v_0.

6 Results

In this last section, we first summarize all obtained results. Recall that a dandelion $D_{m,n}$ consists of a star subgraph S'_m and a path subgraph P'_n.

1. In a path $P_n, n \in \mathbb{N}$, all vertices are solvable, due to the Theorem 2.3.

2. In a star $S_m, m \geq 3$, only the vertex of degree m is solvable, due to the Theorem 2.5.

3. In a starfish $S_{m,n}, m \in \mathbb{N}, n \geq 2$, all vertices are solvable, due to the Theorem 2.9.

4. Generalized $S_{(a_m)}, a_1 \neq \ldots \neq a_m$ is solvable in every vertex, due to the Theorem 2.10.

5. Generalized $S_{(a_m)}, a_1, \ldots, a_m > 1$ is solvable in every vertex, due to the Theorem 2.13.

6. In a dandelion $D_{m,n}, (m,n) \in \mathbb{N}^2$, leaf vertices $s \in V(S'_m)$ are solvable if $n \geq m$ except when $(m,n) \in \{(3,4),(3,5),(4,6)\}$, due to the Theorem 3.7.

7. In a dandelion $D_{m,n}, (m,n) \in \mathbb{N}^2$, all vertices are solvable if $n \geq 2m+3$ except the vertex p_2 if $(m,n) = (4,11)$ and except the vertex p_1 if $(m,n) = (3,10)$, due to the Theorem 3.14.

Finally, we summarize all the problems that have been discussed but remain unsolved.

1. Problem 3.8 asks about leaf vertices $s \in V(S'_m)$ of dandelions when $n < m$.

2. Problem 3.15 asks about non-leaf vertices of dandelions when $n < 2m+3$.

3. Section 2.4, can we characterize $S_{(a_n)}$ containing indices $i \in I$ such that $a_i = 1$?

4. Conjecture 5.1 claims all complete binary trees $T_h, h \geq 4$ are solvable in every vertex.

For discussion, comments on MSE or MO are welcome (see [7, 8, 9, 10]). Likewise, we would like to know if any other class of graphs is being worked on.

References

[1] M. R. Garey, D. S. Johnson, (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, ISBN 0-7167-1045-5.

[2] G. Hamilton, M. Veselovac, *Lazy toad problems*, Mathpickle, 2019, available at https://mathpickle.com/project/lazy-toad-problems-counting-symmetry/.

[3] G. Hamilton, M. Veselovac, *Lazy toads on a star*, Mathpickle, 2019, available at https://mathpickle.com/project/lazy-toads-on-a-star/.

[4] G. Hamilton, Numberphile, *Frog Jumping - Numberphile*, YouTube, 2017, available at https://www.youtube.com/watch?v=X3HDnrehyDM.

[5] M. Krnc, R. Škrekovski, *On the Wiener inverse interval problem, Figure 1.*, MATCH Communications in Mathematical and in Computer Chemistry. **75** (2015), 71–80, available at https://www.researchgate.net/publication/282655921_On_the_Wiener_inverse_interval_problem.

[6] M. Veselovac (Vepir), *Frog problem on a dandelion graph*, Puzzling Stack Exchange, 2020, available at https://puzzling.stackexchange.com/q/105285/32666.

[7] M. Veselovac (Vepir), *Frog problem on leafs of a dandelion graph*, Mathematics Stack Exchange (MSE), 2020, available at https://math.stackexchange.com/q/3937689/318073.

[8] M. Veselovac (Vepir), *Frogs jumping on trees*, Mathematics Stack Exchange (MSE), 2018, available at https://math.stackexchange.com/q/2902660/318073.

[9] M. Veselovac (Vepir), *Is the Frog game solvable in the root of a full binary tree?*, Mathematics Stack Exchange (MSE), 2020, available at https://math.stackexchange.com/q/3800570/318073.

[10] M. Veselovac (Vepir), *Is the Frog game solvable in the root of a full binary tree?*, Math Overflow (MO), 2020, available at https://mathoverflow.net/q/370694/88524.

[11] E. Weisstein, *Centipede Graph*, From MathWorld–A Wolfram Web Resource, available at https://mathworld.wolfram.com/CentipedeGraph.html.

[12] E. Weisstein, *Simple Graph*, From MathWorld–A Wolfram Web Resource, available at https://mathworld.wolfram.com/SimpleGraph.html.