Hanoi Graphs, Sierpiński Graphs, and Sierpiński Triangle Graphs

A REPORT SUBMITTED IN
PARTIAL FULFILMENT OF THE REQUIREMENTS FOR INTERNSHIP
By

Hrishik Koley
Indian Statistical Institute, Bangalore

DURING
01/06/2023 to 31/07/2023

UNDER THE GUIDANCE OF
Dr. Divya Sindhu Lekha

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY KOTTAYAM

1 Introduction

In this report we have discussed the "Tower of Hanoi"(ToH) game and a special variation of the game, called the "Switching Tower of Hanoi"(STH) as well as, the graphs associated to them called Hanoi graph H_p^n and Sierpiński graph S_p^n, respectively. Moreover, we would also discuss a variation of the Sierpiński graphs called the Sierpiński Triangle graphs and it's median. Both the games ToH and STH are aimed at completing a simple task of shifting a stack/pyramid of n discs from one peg to another obeying a set of rules which will be discussed later in sections 2 and 5, respectively.
The Hanoi graph H_p^n consists of all possible arrangements of n discs on p edges in ToH as vertices and if a legal move exists between two given arrangements then the vertices corresponding to these arrangements are connected by an edge. The Sierpiński graph S_p^n consists of all possible arrangements of n discs on p edges in STH as vertices and if a legal move exists between two given arrangements then the vertices corresponding to these arrangements are connected by an edge. We will discuss more about the Hanoi graphs and Sierpiński graphs in sections 3 and 6.
For Hanoi graphs and Sierpinski graphs, we would first define some terminologies related to H_p^n and S_p^n.

- Perfect state of a Tower of Hanoi game is defined as a state in which all the n discs in the game lie on a single peg in such a way that no disc of larger diameter rests on a disc of smaller diameter.
- Regular state of a Tower of Hanoi game is defined as a non-perfect state where the n discs are arranged on the p pegs in such a way that no disc of larger diameter rests on a disc of smaller diameter
- Optimal solution of a Tower of Hanoi game is defined as the set of moves required to move the tower from one peg to another in the minimum number of steps.
- Optimal path of a Tower of Hanoi game is defined as the path containing the sequence of moves involved in the optimal solution for that game.

We would also discuss three different classes of problems associated with this game. These problems entails:

1. The P0 problem refers to the task of shifting the tower from the initial peg to the goal peg in the minimum number of moves.
2. The P1 problem refers to the task of shifting all the discs from an arbitrary initial regular state to a perfect state on the goal peg.
3. The P2 problem refers to the task of shifting all the discs from an arbitrary regular initial state to an arbitrary regular goal state.

The Hanoi graphs and Sierpinski graphs find various applications owing to their fractal structure. Some of these applications are:

- They can be used to study universal topological spaces.
- They can be used to design interconnection networks.

- They can be used to explore various graph properties, such as distance, domination, coloring, embedding, and hamiltonicity.

The non-clique edges of a graph are the edges not present in a complete subgraph of the given graph. Sierpiński Triangle graphs are a variation of Sierpiński graphs formed by contracting all non-clique edges present in the Sierpiński graph. The Sierpiński Triangle graph formed by contracting the non-clique edges of S_p^{n+1} is represented by $\hat{S}_p^n$.

2 Tower of Hanoi

The classical Tower of Hanoi, also known as the Tower of Brahma or Lucas' Tower, is a mathematical game involving 3 pegs and n discs, whose objective is to shift the discs from the initial peg to a different desired goal peg. In the Tower of Hanoi game, a disc can be moved from one peg to any of the other pegs and at any given point of time the discs are always arranged on the pegs in a manner such that the largest disc on the peg lies at the bottom and the diameter of the discs decrease continually as we approach the top of that peg [7].

The rules of the Tower of Hanoi game are as follows [7]:

1. In each move the player must move exactly one disc from a peg and shift to some other peg.
2. In each move the player is only allowed to move the disc on the top of a stack on a peg.
3. In each move the player is only allowed to place the removed disc on top a disc of larger diameter, that is, at no given point of time should there be a larger disc resting on top of a smaller disc.

Theorem 2.0.1 (Theorem 2.1 in [4])**.** *The classical Tower of Hanoi task for n discs ($n \in \mathbb{N}$) has a unique optimal solution of length* $2^n - 1$.

Proof. We use induction to prove the above theorem.
Base Case: For n=1, we can move the n discs from initial peg to the goal peg in just 1 move, which is equal to $2^n - 1$ for n=1.
Induction Hypothesis: Let for n=k, the number of moves required to reach perfect goal state be $2^k - 1$.
Inductive Step: Let n=k+1. For this case, the optimal solution entails shifting the smallest k discs from the initial peg to the intermediate peg (the peg other than the initial and the goal peg), which takes $2^k - 1$ moves optimally. Then the largest disc is moved from initial peg to goal peg, which can be done in 1 move. Finally we move the stack of k discs from the intermediate peg to the top of the largest disc on the goal peg, which takes $2^k - 1$ moves optimally. Thus, the total number of moves required in shifting the entire stack of k+1 discs from the initial peg to the goal peg optimally requires ($2^k - 1$ + 1 + $2^k - 1$)=$2^{k+1} - 1$ moves.
Hence, we conclude that in the classical Tower of Hanoi task for n discs has a unique optimal solution of length $2^n - 1$. □

The Tower of Hanoi problem has many variations, such as, the Linear Tower of Hanoi problem, the Twin Tower problem, the Switching Tower of Hanoi problem, and many others. We will discuss about the Switching Tower of Hanoi in section 5.

3 Hanoi Graphs and their properties

Hanoi graphs, H_p^n, can be defined as the graphs associated with ToH problem consisting of p pegs and n discs. The vertices of H_p^n are labelled as $s = k_1k_2...k_n$, where k_i refers to the peg on which the i^{th} disc rests. The vertex set of the graph H_p^n is defined as, V(H_p^n)=$\{s = k_1k_2...k_n : 0 \leq k_i \leq p-1, \forall 1 \leq i \leq n\}$. The number of elements in V(H_p^n) is equal to the total number of possible arrangements of the n discs on the p pegs, that is, equal to p^n. The edge set of H_p^n is represented by E(H_p^n), and an edge exists between 2 given vertices, represented by s_1 and s_2, if and only if state s_1 can be achieved from state s_2 and vice versa by exactly one legal move. The edge set of H_3^n is defined as, E(H_3^n)=$\{\{s_d i(3-i-j)^d - 1, s_d i(3-i-j)^d - 1\}|$
$i, j \in T, i \neq j, d \in [n], s_d \in T^{n-d}\}$, where $T = 0, 1, 2$ and T^m is the set of all possible $m - tuples$ where the elements of a $m - tuple$ can take values belonging to T.

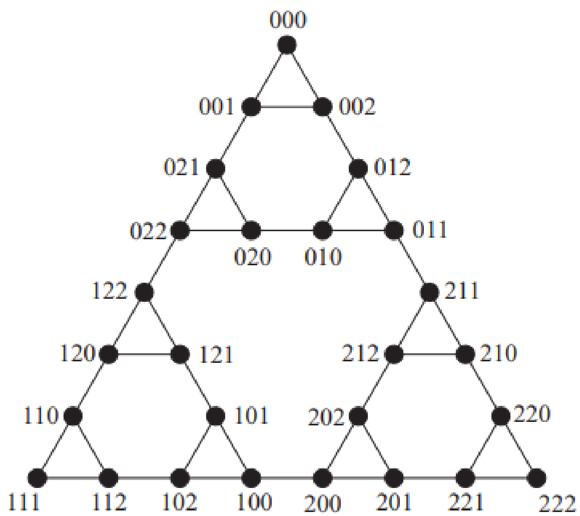

Figure 1: Hanoi graph H_3^3

3.1 Properties of the Hanoi Graph H_3^n

Here, we discuss the properties of Hanoi graph H_3^n. First we start by defining some graph theoretic and group theoretic terms. [5][6]

- $\delta(G)$ is defined as the minimum degree of a vertex in graph G
- $\chi(G)$ is the chromatic number of graph G defined as the least number of colours required to colour the vertices of the graph in such a way that no two adjacent vertices share the same colour.
- $\kappa(G)$ is the connectivity of graph G which is defined as the minimal vertex cut. A vertex cut of a connected graph G is the set of vertices the removal of which renders G disconnected.
- $d(v)$ is the total distance of a vertex v in the graph G which is defined as, $d(v) = \sum_{u \in V} d(u, v)$

- $\overline{d}(v)$ is the average distance of a vertex v in the graph G which is defined as $\overline{d}(v) = \frac{d(v)}{|G|-1}$.

- $\epsilon(v)$ is the eccentricity of a vertex v in the graph G which is defined as, $\epsilon(v) = max_{u \in V} d(u,v)$.

- $prox(G)$ is the proximity of graph G which is defined as $prox(G) = min\{\overline{d}(v)|v \in V\}$.

- $rem(G)$ is the remoteness of graph G which is defined as $rem(G) = max\{\overline{d}(v)|v \in V\}$.

- $diam(G)$ is the diameter of the graph G which is defined as $diam(G) = max\{\epsilon(v)|v \in V\}$.

- $rad(G)$ is the radius of the graph G which is defined as $rad(G) = min\{\epsilon(v)|v \in V\}$.

- $N[D]$ is the neighbourhood of the set of vertices D, which includes all those vertices that are adjacent to at least one vertex in set D.

- $\gamma(G)$ is the domination number of graph G is the minimum size of a G-dominating set, that is, the size of the smallest subset D of $V(G)$ such that $N[D] = V(G)$

- We call set $C \in V(G)$ to be 1-error correcting if the neighbourhood of the elements of C do not overlap.

- A permutation group is defined as a group G whose elements are permutations of a given set T and whose group operation is the composition of permutations in G. The group of all permutations defined on a set T is the symmetric group, represented by Sym(T).

- A group isomorphism is a function between two groups that sets up a one-to-one correspondence between the elements of the groups with respect to the given group operations.

- The automorphism group on a group X, represented by Aut(X), is defined as the group of all group isomorphisms from X to itself.

Now we state some graph theoretic theorems.

Theorem 3.1.1 (Subsection 1.1.2 in [5]). *For any graph G, $\kappa(G) \leq \delta(G)$.*

Theorem 3.1.2 (Theorem 1.40 in [5]). *(Kuratowski's Theorem) A graph G is planar if and only if it contains no subdivision of $K_{3,3}$ or K_5.*

Theorem 3.1.3. *[4] If $C \in V(G)$ is 1-error correcting, then $\gamma(G) \geq |C|$.*

Corollary 3.1.3.1. *[4] If C is a perfect code of G, then $\gamma(G) = |C|$.*

Now we will state the properties of the Hanoi graph H_3^n.

- H_3^n is a Hamiltonian graph.[4]

- H_3^n is a planar graph.[4] This is because the maximum degree of a vertex in H_3^n is 3 so it cannot have a subdivison K_5 and atleast two adjacent vertices of a given vertex are connected by an edge so we can say that H_3^n does not have a subdivision of $K_{3,3}$. Thus, by Theorem 3.1.2., we can conclude that H_3^n is a planar graph.

- $\chi(H_3^n) = 3$. [Proposition 2.21 in [4]] This is because for a given $s \in T^n$ if we define $f(s) = \sum_{i \in T} k_i (mod 3)$, then we can observe that for any two adjacent vertices, s_1 and s_2, the values of $f(s_1)$ and $f(s_2)$ are never equal. So, we colour each vertex with equal value of $f(s)$ with the same colour. The values that $f(s)$ can take are 0,1, and 2. So, we need only 3 colours.

- $\kappa(H_3^n) = 2$. [Proposition 2.22 in [4]] This is because as H_3^n is a Hamiltonian graph so must be k-connected where $k \geq 2$ and as $\delta(H_3^n) = 2$ so from Theorem 3.1.1., we know that, $\kappa(H_3^n) \leq 2$. Combining these two facts we get $\kappa(H_3^n) = 2$.

- $\gamma(H_3^n) = \frac{1}{4}(3^n + 2 + (-1)^n)$.[4] As H_3^n has a perfect code containing $\frac{1}{4}(3^n + 2 + (-1)^n)$ codewords, so by Corollary 3.1.3.1., we can conclude that $\gamma(H_3^n) = \frac{1}{4}(3^n + 2 + (-1)^n)$.

- $rad(H_3^n) = 2^n - 2^{n-2}$.[6]

- $diam(H_3^n) = 2^n - 1$.[Theorem 2.25 in [4]] This can be proved by considering a H_3^{m+1} and assuming the induction hypothesis that $diam(H_1^m) = 2_1^m$. Now we consider $s, t \in T^m$ and we know that $d(is, jt) \leq d(is, ik^m) + 1 + d(jk^m, jt) \leq 2^m - 1 + 1 + 2^m - 1 = 2^{m+1} - 1$ (from induction hypothesis). Hence, we can conclude from the inductive proof that $diam(H_3^n) = 2^n - 1$.

- $prox(H_3^n) = \frac{d_2(n)}{3^n - 1}$. [Hypothesis A in [6]] $d_2(n) = min\{d(s)|s \in T^n\}, \forall n \in \mathbb{N}$.

- $rem(H_3^n) = \frac{d_0(n)}{3^n - 1}$. [Proposition 1 in [6]] $d_0(n) = max\{d(s)|s \in T^n\}, \forall n \in \mathbb{N}$.

- For any $n \in \mathbb{N}, Aut(H_3^n) \cong Sym(T)$. [Theorem 2.23 in [4]] The six automorphisms of H_3^n, $g_\sigma : T^n \to T^n, s \to \sigma(s_n)...\sigma(s_1)$ for $\sigma \in Sym(T)$, form a subgroup of $Aut(H_3^n)$ isomorphic to $Sym(T)$. Now let, $g \in Aut(H_3^n)$. Since g preserves degrees, we necessarily have $g(k^n) = \sigma(k)^n$ for some $\sigma \in Sym(T)$ and $\forall k \in T$.

4 Classical Problems related to Hanoi Graphs

Now, we will be discussing three classes of classical problems, P0,P1, and P2, related to Hanoi graphs H_p^n.

4.1 Perfect to Perfect or P0 problem

The P0 problem on H_3^n entails the task of shifting the entire tower of n discs from one peg to another. Before diving into the ways of solving such a problem we first state some observations and theorems.

Theorem 4.1.1 (Proposition 2.2 in [4]). *In the optimal solution to transfer $n \in \mathbb{N}$ discs from one peg to another, disc $d \in [n]$ moves for the first time in step 2^{d-1} and for the last time in step $2^n - 2^{d-1}$; in particular, the largest disc n moves exactly once, namely in the middle of the solution.*

Proof. In the optimal solution, the strategy is to move a the tower of d-1 discs comprising of discs $\{1, 2, 3, ..., d-1\}$ to the goal peg (if parity of n and d are different) or to the intermediate peg (if parity of n and d are same) and then move the d^{th} disc to the intermediate peg (if parity of n and d are different) or to the goal peg (if parity of n and d are same) and then the tower of those d-1 discs is shifted and placed over the d^{th} disc. In optimal solution, the number of moves required to shift a tower of d-1 discs from one peg to another is $2^{d-1}-1$. So, the d^{th} disc is moved for the first time in the $(2^{d-1})^{th}$ move. Moreover, as total number of moves in moving the n discs is $2^n - 1$ so, by symmetry we know that the d^{th} disc is moved for the last time in the $(2^n - 2^{d-1})^{th}$ move. □

Remark 4.1.1. [4] In an optimal solution of moving a tower from a peg to another peg the smallest disc is shifted in every odd move.

First we look at Olive's Algorithm[4] to solve the P0 problem which makes use of Remark 4.1.1.

Algorithm 1 Olive's algorithm

```
Require: n: number of discs {n ∈ ℕ}
Require: i: source peg {i ∈ T}
Require: j: goal peg {j ∈ T}
  if n=0 or i=j then
     STOP
  end if
  if n is odd then
     move disc 1 from peg i to peg j
  else
     move disc 1 from peg i to peg 3-i-j
  end if
  remember move direction of disc
  while not all discs are on peg j do
     make legal move of disc not equal 1
     make one move of disc 1 cyclically in its proper direction
  end while
```

Now we take a look at the Idle Peg Algorithm[4] which makes use of Theorem 4.1.1 to determine the idle peg, that is, the peg not involved in a given move. It uses the idea of this idle peg to make the moves and reach the perfect state using the optimal solution.

Algorithm 2 Idle peg algorithm

```
Require: n: number of discs {n ∈ ℕ}
Require: i: source peg {i ∈ T}
Require: j: goal peg {j ∈ T}
  idle ← i
  dir ← (−1)^n (j − i)
  while not all discs are on peg j do
     idle ← (idle + dir) mod 3
     make legal move between pegs different from idle
  end while
```

We can also make use of a recursive approach[4]to solve the P0 problem, that is, we make use of the solution of the P0 problem for the graph H_3^{n-1} to formulate a solution of the P0 problem for the graph H_3^n.

Algorithm 3 Recursive algorithm

```
  Procedure p0(n,i,j)
Require: n: number of discs {n ∈ ℕ}
Require: i: source peg {i ∈ T}
Require: j: goal peg {j ∈ T}
  if n ≠ 0 and i ≠ j then
     k ← 3 − i − j                      ▷ the auxiliary peg different from i and j
     p0(n-1, i, k)                      ▷ transfers n-1 smallest discs to auxiliary peg
     move disc n from peg i to peg j    ▷ moves largest disc to goal peg
     p0(n-1, k, j)                      ▷ transfers n-1 smallest discs to goal peg
  end if
```

4.2 Regular to Perfect or P1 problem

The P1 problem on H_3^n entails the task of moving the n discs from a regular state to a perfect state on a given peg. Before diving into ways of solving the P1 problem we need to first state some theorems.

Theorem 4.2.1 (Proposition 2.3 in [4]). *A state $s \in T^n$ belongs to the optimal solution path from i^n to j^n if and only if s is admissible, that is, the corresponding arrangements on all three pegs are admissible.*

Proof. We prove by induction that there are 2^n admissible states in T^n with respect to i and j and that all 2^n states on the shortest path from i^n to j^n are admissible.
For n = 0, we just note that the empty arrangement on a peg is admissible.
Let $\overline{s} = s_{n-1}...s_1$. Then $s = s_n\overline{s}$ is admissible with respect to i and j if and only if $s_n \in \{i, j\}$ and $\overline{s}$ is admissible in T^{n-1} with respect to s_n and 3-i-j. Therefore, by induction assumption, there are $2 \cdot 2^{n-1} = 2^n$ admissible states in T^n. The first 2^{n-1} states on the optimal path from i^n to j^n are those where $s_n = i$, that is, the largest disc n is lying on a disc of different parity, and $\overline{s}$ belongs to the shortest path from $i^{n-1}$to $(3-i-j)^{n-1}$ such that s is admissible with respect to i and j because disc n replaces the immovable disc n + 1. The same argument applies for the last 2^{n-1} states, where now disc n is on disc n + 3 of opposite parity. □

Theorem 4.2.2. *[4] For $j \in [p]$ and any vertex s=$s_n...s_1$ of H_p^n,*
$d(s, j^n) = \sum_{i=1}^{n}(s_d \neq j).2^{d-1}$

Proof. By induction on n. The statement is trivial for n = 0. Let $n \in \mathbb{N}$ and $s = s_{n+1}\overline{s}$, $\overline{s} \in [p]_0^n$.
If $s_{n+1} = j$, then we can use the shortest path in H_p^n from $\overline{s}$ to j^n and add a j in front of each vertex. Hence
$d(s, j^{n+1}) \leq \sum_{d=1}^{n}(s_d \neq j).2^{d-1} = \sum_{d=1}^{n+1}(s_d \neq j).2^{d-1}$.
If $s_{n+1} \neq j$, we can compose a path from s to j^{n+1} by going from $s_{n+1}\overline{s}$ to $s_{n+1}k^n$ on a shortest path of length $\leq \sum_{d=1}^{n}(s_d \neq j).2^{d-1}$, and then we move to jk^n and finally from there to j^{n+1} in $2^n - 1$ steps, altogether
$d(s, j^{n+1}) \leq \sum_{d=1}^{n+1}(s_d \neq j).2^{d-1}$. □

Theorem 4.2.3 (Theorem 2.7 in [4]). *The task to get from a regular state $s \in T^n$ to the perfect one on peg $j \in T$ has a unique optimal solution of length $\leq 2^n - 1$.*

Proof. We use induction to prove this theorem.
Base Case: If n=1, then we can move from regular state s to perfect state in $\leq 2^n - 1$ moves.
Induction Hypothesis: Let $\forall\ n \leq m$, the theorem holds true.
Inductive Step: The regular state s belongs to any one of the 3 subgraphs iS_3^m, jS_3^m, or kS_3^m. It takes $\leq 2^m - 1$ moves to go from regular state s to an extreme vertex in the subgraph, then we make another move to shift to one of the other two subgraphs if needed, and lastly we move to the extreme vertex of S_3^{m+1} in $2^m - 1$. Thus, we take $\leq (2^m - 1 + 1 + 2^m - 1) = 2^{m+1} - 1$ moves to go from regular state s to perfect state. □

Theorem 4.2.4 (Proposition 2.9 in [4]). *Suppose that in an optimal solution for a P1 task if disc d is moved for $d \neq 1$ then in the next move, disc 1 is moved and it is moved onto disc d if and only if d is even.*

Proof. If a disc $d \neq 1$ is moved onto another disc then after that there are only 3 possible legal moves moving the disc d back to it's last position or moving disc 1 to any of the other two pegs. Thus, to make progress in the game we need to move disc 1.
For n = 2, disc 1 has to move atop disc 2 after the latter's only move. For the induction step we may assume that disc n + 1 is originally not on its goal peg, in which case it moves exactly once. Before this move the induction assumption applies. The move of disc n + 1 is followed by a transfer of a perfect n-tower onto disc n + 1 and here, according to Olive's solution, the first move, necessarily by disc 1, is to the goal peg if and only if n is odd, i.e. n + 1 is even. □

For a solution to the P1 problem, we need to first decide if the regular state s belongs to the optimal path for which we use the following algorithm[4] which makes use of Theorem 4.2.1.

Algorithm 4 Detection of deviation from optimal path

Require: n: number of discs $\{n \in \mathbb{N}\}$
Require: i: source peg $\{i \in T\}$
Require: j: goal peg $\{j \in T\}$
Require: s: regular state on optimal path $\{s \in T^n\}$
  $k \leftarrow 3 - i - j$ ▷ state of the P1-automaton
  while $d \geq 2$ **do**
    if $s_d = k$ **then**
      s is not on optimal path, STOP
    else
      $k \leftarrow 3 - k - s_d$ ▷ updated state of P1-automaton
    end if
    $d \leftarrow d - 1$
  end while
  if $s_1 = k$ **then** s is not on optimal path, STOP
  end if
  s is on optimal path

Now there can be two possible approaches. Let us first consider the approach for the case where s is on optimal path.[4] Before stating the algorithm we first state what we mean by right direction for a disc. We consider the following cases for stating the right direction:

- If n=even and disc d to be moved is odd then the right direction is: initial peg → intermediate peg → goal peg → initial peg
- If n=even and disc d to be moved is even then the right direction is: initial peg → goal peg → intermediate peg → initial peg
- If n=odd and disc d to be moved is even then the right direction is: initial peg → intermediate peg → goal peg → initial peg
- If n=odd and disc d to be moved is odd then the right direction is: initial peg → goal peg → intermediate peg → initial peg

Algorithm 5 Optimal first move if s is on optimal path

Require: n: number of discs $\{n \in \mathbb{N}\}$
Require: i: source peg $\{i \in T\}$
Require: j: goal peg $\{j \in T\}$
Require: s: regular state on optimal path $\{s \in T^n\}$
  let d be the smallest top disc of s different from 1
  if the legal move of d is in the right direction **then** move disc d
  elsemove disc 1 in the right direction
  end if

Next we state the algorithm to follow if s is not on optimal path.[4]

Algorithm 6 Best first move if s is not on optimal path

  Procedure : p1(n,j,s)
Require: n: number of discs $\{n \in \mathbb{N}\}$
Require: j: goal peg $\{j \in T\}$
Require: s: regular state on optimal path $\{s \in T^n\}$
  $\mu \leftarrow 0$ ▷ length of path
  $\delta \leftarrow n + 1$ ▷ active disc
  $k \leftarrow j$ ▷ state of P1-automaton
  while $d \geq 1$ **do**
    if **then**$s_d \neq k$
      $\mu \leftarrow \mu + 2^{d-1}$
      $\delta \leftarrow d$
      $k \leftarrow 3 - k - s_d$ ▷ updated state of P1-automaton
    end if
    $d \leftarrow d - 1$
  end while

4.3 Regular to Regular or P2 problem

The P2 problem on H_3^n entails the task of shifting the n discs from a regular state to another regular state. Before diving into ways of solving the P2 problem we need to first state some theorems.

Theorem 4.3.1. *[4] The maximum number of moves required to go from regular state $\overline{s}$ to regular state $\overline{t}$ is $2^n - 1$.*

Proof. Let, $i \neq j \neq k$ and $s, t \in T^n$
$d(is, jt) \leq d(is, ik^n) + 1 + d(jk^n, jt) = d(s, k^n) + 1 + d(k^n, t) \leq (2^n - 1) + 1 + (2^n - 1) = 2^{n+1} - 1$ (Using Theorem 4.2.3) □

Now we look at the algorithm to solve a P2 problem for a H_3^n.[4]

Algorithm 7 Algorithm to solve P2 problem for H_3^n

Require: n: number of discs $\{n \in \mathbb{N}\}$
Require: s: regular initial state $\{s \in T^n\}$
Require: t: regular goal state $\{t \in T^n\}$
  $\mu \leftarrow 0$ ▷ length of path
  $N \leftarrow n$ ▷ largest disc to be moved
  while $s_N = t_N$ **do**
    $N \leftarrow N - 1$
  end while
  if N=0 **then**
    STOP
  end if
  $C \leftarrow 1$ ▷ case of P2 decision
  $\overline{s} \leftarrow s_{N-1}...s_1$, $\overline{t} \leftarrow t_{N-1}...t_1$ ▷ $s = s_n...s_1$ and $t = t_n...t_1$
  $p1(N-1, \overline{s}, 3 - s_N - t_N)$
  $\mu_1 \leftarrow \mu, \delta_1 \leftarrow \delta, i_1 \leftarrow i$ ▷ μ, δ, i : output of p1(n,j,s)
  $p1(N-1, \overline{t}, 3 - s_N - t_N)$
  $\mu_1 \leftarrow \mu_1 + 1 + \mu$ ▷ length of one-move path
  $p1(N-1, \overline{s}, t_N)$
  $\mu_2 \leftarrow \mu, \delta_2 \leftarrow \delta, i_2 \leftarrow i$ ▷ μ, δ, i : output of p1(n,j,s)
  $p1(N-1, \overline{t}, s_N)$
  $\mu_2 \leftarrow \mu_2 + 2^{N-1} + 1 + \mu$ ▷ length of two-move path
  $\mu \leftarrow \mu_1$
  if $\mu_1 \geq \mu_2$ **then**
    $C \leftarrow C + 1$, $\mu \leftarrow \mu_2$
  end if
  if $\mu_1 = \mu_2$ **then**
    $C \leftarrow C + 1$
  end if

5 Switching Tower of Hanoi

Switching Tower of Hanoi (STH) is one of the many variations of the Tower of Hanoi (ToH) game. The goal of STH is the same as the goal of ToH, that is, we need to shift the tower from one peg to another. However, STH has a modified set of rules, which can be stated as follows[4]:

1. A disc can only be placed on top a disc of larger diameter than itself.
2. The Switching Tower of Hanoi allows 2 types of legal moves:
   (a) The smallest disc can be shifted from it's initial peg to any other peg.
   (b) Any disc (say m) other than the smallest one lying on top of a stack on a peg can be switched with a tower lying on another peg if and only if the tower consists of all the discs smaller than disc m.

6 Sierpiński Graphs and their properties

Sierpiński graphs are extensively studied graphs of fractal nature with applications in topology, mathematics of ToH and computer science. In context of the STH, it is used to map every possible arrangement in the game in a graphical manner where the vertices correspond to the arrangements in the game and the edges connects two vertices if one can be reached from the other by exactly one legal move.
For moving from a perfect state on the initial peg to a perfect state on the goal peg, STH takes $2^n - 1$ moves in the optimal solution when there are n discs and 3 pegs involved. The Sierpiński graph over n discs and p pegs is represented by S_p^n. The vertex set of S_p^n is defined as $V(S_p^n) = \{s = k_1k_2...k_n : 0 \leq k_i \leq p-1, \forall 1 \leq i \leq n\}$. The edge set of S_p^n is defined as $E(S_p^n) = \{\{sij^{d-1}, sji^{d-1}\} | 0 \leq i,j \leq p-1, i \neq j, 1 \leq d \leq n, s \in T^{n-d}, T = \{0,1,...,p-1\}\}$.[4]

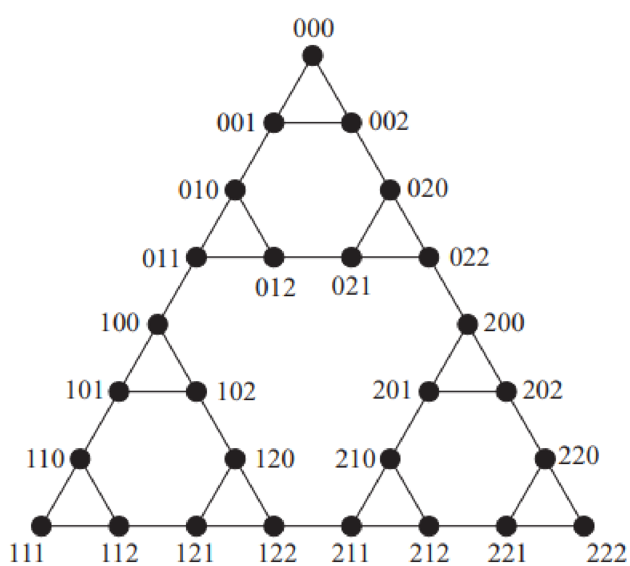

Figure 2: Sierpiński graph S_3^3

6.1 Properties of small Sierpiński graphs

While considering the small Sierpiński graphs we will be mainly focusing on the Sierpiński graphs, S_1^n, S_p^1, S_2^n, and S_3^n. Let, us look at them one by one.

- First we consider the Sierpiński graph S_1^n. We can observe that in this graph p=1, that is, it has only one peg, and so all the discs lie on this peg. Therefore, this is a simple graph with a single vertex.[6]

- Next we consider the Sierpiński graph S_p^1. Here, we have n=1 and so the vertices are just the peg on which the disc lies. Therefore, this is a simple graph with p vertices and the disc can shift from any peg to the other and so all the vertices are interconnected. Thus, the graph S_p^1 is a complete graph of order p, that is, $S_n^1 \cong K_p$.[6]

- Now we consider the Sierpiński graph S_2^n. The graph S_2^n is isomorphic to a path of length 2^n, that is, $S_2^n \cong P_{2^n}$. Thus, we first consider the properties of graph P_{1+k}. We first express the vertex set of P_{1+k} as, $V(P_{1+k}) = \{0, 1, 2, ..., k\}$, where an edge exists between two vertices if the two numbers corresponding to those vertices are adjacent on the number line. We can observe that S_2^n is planar. It can also be trivially observed that radius of P_{1+k} is, $rad(P_{1+k}) = \frac{1}{2}$(k+(k mod 2)), and it's diameter is, $diam(P_{1+k}) = k$. It can also be easily observed that for any vertex $v \in V(P_{1+k})$, $d(v) = \frac{1}{2}$k(k+1)-v(k-v), and therefore, $\overline{d}(v) = \frac{k+1}{2} - \frac{v(k-v)}{k}$. Thus, we can observe that the proximity of P_{k+1} is, $prox(P_{k+1}) = \frac{1}{4k}(k^2$ + 2k + (k mod 2)) and the remoteness of P_{k+1} is, $rem(P_{k+1}) = \frac{k+1}{2}$.[6]

- Lastly we consider the Sierpiński graph S_3^n. This graph is isomorphic to the Hanoi graph with n discs and 3 pegs, that is, $S_3^n \cong H_3^n$. Therefore, S_3^n has all the properties of H_3^n stated in section 3.1.[6]

6.2 Properties of general Sierpiński graphs

Now we will state some properties of the general Sierpiński graph S_p^n.

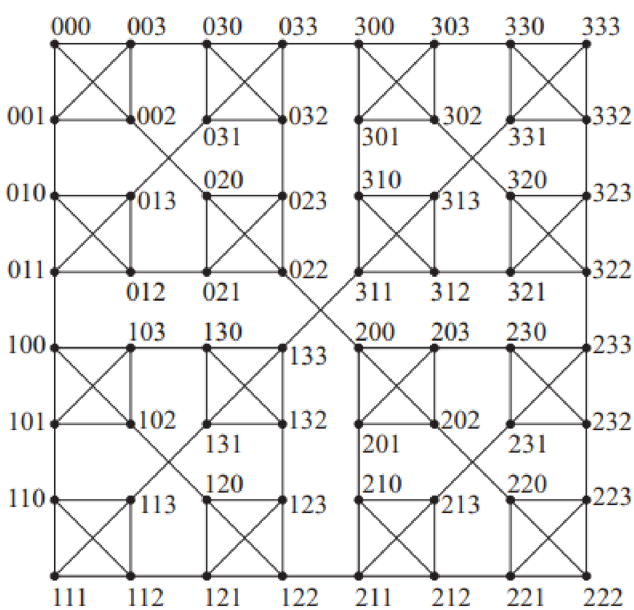

Figure 3: Sierpiński graph S_4^3

- For any $p \geq 2$ and any $n \in \mathbb{N}$, $diam(S_p^n) = 2^n - 1$.[4]
- For any $n \geq 1$ and any $p \geq 3$, the graph S_p^n is hamiltonian.[Proposition 4.12 in [4]]
- The only Sierpiński graphs that are planar are S_3^n, S_2^n, S_1^n, S_p^0, S_4^1, and S_4^2. We know that S_p^{n-1} is a subgraph of S_p^n, and as for $p \geq 5$ and S_p^1 is non-planar by Kuratowski's theorem as it has a K_5 subdivision, so by induction we can say that for $p \geq 5$, all graphs S_p^n are non-planar when $n \geq 1$. In S_4^3 when we try to deaw the overlapping diagonals outwardly the overlap instead with the edges joining any two S_4^2 subgraphs, so it is non-planar and so all other S_4^n.[4]
- We can express the edge set of S_p^{n+1} recursively as:
  $\forall n \in \mathbb{N} : E(S_p^{n+1}) = \{\{ir, is\} | i \in [p]_0, \{r, s\} \in E(S_p^n)\} \cup \{\{ij^n, ji^n\} | i, j \in [p]_0, i \neq j\}$[4]
- For any $n, p \in \mathbb{N}$, $Aut(S_p^n) \cong Sym([p]_0)$.[Theorem 4.14 in [4]] We have already seen that $Aut(S_3^n) \cong Sym(T)$, which holds as $S_3^n \cong H_3^n$. We can extend this to all S_p^n for $p \geq 3$.
- The clique number of S_p^n is, $\omega(S_p^n) = p$.[Theorem 4.3 in [4]] For $n \in \mathbb{N}$, each S_p^n contains p^{n-1} isomorphic copies of $S_p^1 \cong K_p$, namely the subgraphs $\underline{s}S_p^1$ with $s \in [p]_0^{n-1}$. The constitute the p-cliques in S_p^n and they are the maximal cliques in S_p^n as the degree of all but the extreme vertices is p and one of the edges from each of these vertices connect them to another subgraph of S_p^n.

7 Classical Problems related to Sierpiński Graphs

In context of the Sierpiński graphs, the P0 problem is considered rather important. So, we will be stating algorithms for the P0 problem only.

7.1 Perfect to Perfect or P0 problem

Before discussing about the P0 problem for Sierpiński graphs we first take a look at some theorems.

Theorem 7.1.1 (Proposition 4.5 in [4])**.** *For $j \in [p]$ and any vertex $s{=}s_n...s_1$ of S_p^n,* $d(s, j^n) = \sum_{i=1}^{n}(s_d \neq j).2^{d-1}$

Proof. We use induction to prove this theorem. The statement is trivial for n = 0. Let $n \in \mathbb{N}$ and $s = s_{n+1}\overline{s}$, $\overline{s} \in [p]_0^n$.
If $s_{n+1} = j$, then we can use the shortest path in S_p^n from $\overline{s}$ to j^n and add a j in front of each vertex. Hence
$d(s, j^{n+1}) \leq \sum_{d=1}^{n}(s_d \neq j).2^{d-1} = \sum_{d=1}^{n+1}(s_d \neq j).2^{d-1}$.
If $s_{n+1} \neq j$, we can compose a path from s to j^{n+1} by going from $s_{n+1}\overline{s}$ to $s_{n+1}j^n$ on a shortest path of length $\leq \sum_{d=1}^{n}(s_d \neq j).2^{d-1}$, and then we move to js_{n+1}^n and finally from there to j^{n+1} in $2^n - 1$ steps, altogether
$d(s, j^{n+1}) \leq \sum_{d=1}^{n+1}(s_d \neq j).2^{d-1}$. □

Theorem 7.1.2. *[4] In a Sierpiński graph S_3^n, the number of moves involved in an optimal solution for shifting the tower from one peg to another is $2^n - 1$.*

Proof. The goal peg and initial peg are different. So, we can use Theorem 7.1.1 and write the expression as:
$d(i^n, j^n) = \sum_{i=1}^{n} 2^{d-1} = 2^n - 1.$ □

Remark 7.1.1. [4] We can observe that in an optimal solution of the P0 problem for Sierpiński graph, disc 1 is moved in every odd move and a switch is made in every even move.

Now we will be looking at some algorithms to solve the P0 task for the Sierpiński graph S_3^n. We make use of Remark 7.1.1 to devise the following algorithm.

Algorithm 8 Algorithm to solve P0 task for S_3^n

```
Require: n: number of discs {n ∈ ℕ}
Require: i: source peg {i ∈ T}
Require: j: goal peg {j ∈ T}
  if n=0 or i=j then
     STOP
  end if
  move disc 1 to goal peg
  while not all discs are on peg j do
     switch disc m on top of peg i with tower of m-1 discs on peg j
     move disc 1 to goal peg
  end while
```

We now devise a algorithm using a recursive approach to solve the P0 problem for S_3^n.[4]

Algorithm 9 Recursive algorithm

```
  Procedure p0s(n,i,j)
Require: n: number of discs {n ∈ ℕ}
Require: i: source peg {i ∈ T}
Require: j: goal peg {j ∈ T}
  if n ≠ 0 and i ≠ j then
     p0s(n-1, i, j)          ▷ transfers n-1 smallest discs to goal peg to give state ij^(n-1)
     switch disc n with tower of n-1 discs          ▷ switch is done to reach state ji^(n-1)
     p0s(n-1, i, j)          ▷ transfers n-1 smallest discs to goal peg to reach state j^n
  end if
```

8 Sierpiński-type Graphs

We first start by indroducing the different variations of Sierpiński graphs. We come across 4 main variations of the Sierpiński graphs, namely, Sierpiński Triangle Graphs, Schreier Graphs, WK-recursive Networks, and Regularizations.[2]

1. **Sierpiński Triangle Graphs :** The class of Sierpiński Triangle Graphs $\hat{S}_p^n$ is obtained by contracting all the non-clique edges of the Sierpiński Graph S_p^{n+1}.
2. **Schreier Graphs :** The class of Schreier Graphs $H^{(n)}$ is obtained by adding loops on the extreme vertices of the Sierpiński Graph S_3^n.

3. **WK-recursive Networks :** The class of WK-recursive Networks $WK(p,n)$ is obtained by adding p additional open edges to each of the extreme vertices of S_p^n. It is used as a model for interconnection networks.

4. **Regularizations :** Regularizations are broadly classified into two types, ${}^{+}S_p^n$ and ${}^{++}S_p^n$.

   (a) ${}^{+}S_p^n$ *:* These type of regularizations are obtained by adding an additional vertex w and joining it to all the extreme vertices of S_p^n.

   (b) ${}^{++}S_p^n$: These type of regularizations are obtained by adding a copy of S_p^{n-1} and joining all of it's extreme vertices with the extreme vertices of S_p^n.

9 Sierpiński Triangle Graphs

We will be diving deep into the details and properties of just one variation of Sierpiński graphs, that is, the Sierpiński triangle graphs. We begin by defining the set $\hat{P}$ for $\hat{S}_p^n$ as, $\hat{P} = \{\hat{0}, \hat{1}, ..., p \hat{-} 1\}$.[1]
Now we move on to define the vertex set and the edge set of $\hat{S}_p^n$. We express the vertex set of $\hat{S}_p^n$ as V($\hat{S}_p^n$) and define it as, $V(\hat{S}_p^n) = \hat{P} \bigcup \{\underline{s}\hat{ij} | \underline{s} \in P^{v-1}, v \in [n], i,j \in^P C_2\}$. The edge set is expressed as E($\hat{S}_p^n$) and defined as $E(\hat{S}_p^{n+1}) = \{\{\hat{k}, k^n \hat{jk}\} | k \in P; j \in P\{k\}\} \bigcup \{\{\underline{s}\hat{ij}, \underline{s}\hat{ik}\} | \underline{s} \in P^n; i \in P;$
$j,k \in^{P-\{i\}} C_2\} \bigcup \{\{\underline{s}ki^{n-v}\hat{ij}, \underline{s}\hat{ik}\} | \underline{s} \in P^{v-1}; v \in [n]; i \in P; j,k \in P - \{i\}\}$.[1]

The number of vertices in $\hat{S}_p^n$ is represented by $|\hat{S}_p^n|$ and we get that[1]:
$\hat{S}_p^n = |S_p^{n+1}| - \frac{|S_p^{n+1}|-p}{2}$
$=p^{n+1} - \frac{p^{n+1}-p}{2}$
$=\frac{p^{n+1}+p}{2}$
$=\frac{p}{2}(p^n + 1)$
The number of edges in $\hat{S}_p^n$ is represented by $||\hat{S}_p^n||$ and we get that[1]:
$||\hat{S}_p^n||$=(number of unit p-gons in S_p^{n+1}) $\cdot^P C_2$
$={}^P C_2 \cdot p^n$
$=\frac{p-1}{2}(p^{n+1})$
Now we define the 3 types of vertices in $\hat{S}_p^n$[1]:

1. **Primitive Vertices :** The extreme vertices of $\hat{S}_p^n$, represented by $\hat{\mathrm{k}}$ where $k \in P$, are called the primitive vertices of $\hat{S}_p^n$. The primitive vertices are p in number.

2. **Critical Vertices :** Moves of the largest disc n+1 between pegs i and j in Switching Tower of Hanoi are reflected by vertex $\hat{ij}$ in $\hat{S}_p^n$ and are called critical vertices of $\hat{S}_p^n$. The critical vertices are ${}^pC_2$ in number.

3. **Generic Vertices :** All vertices other than the primitive and critical vertices in $\hat{S}_p^n$ are called generic vertices of $\hat{S}_p^n$. The generic vertices are $\frac{p}{2}(p^n - 1) - {}^pC_2$ in number.

There are generally three types of edges in the graph $\hat{S}_3^{n+1}$.[1]

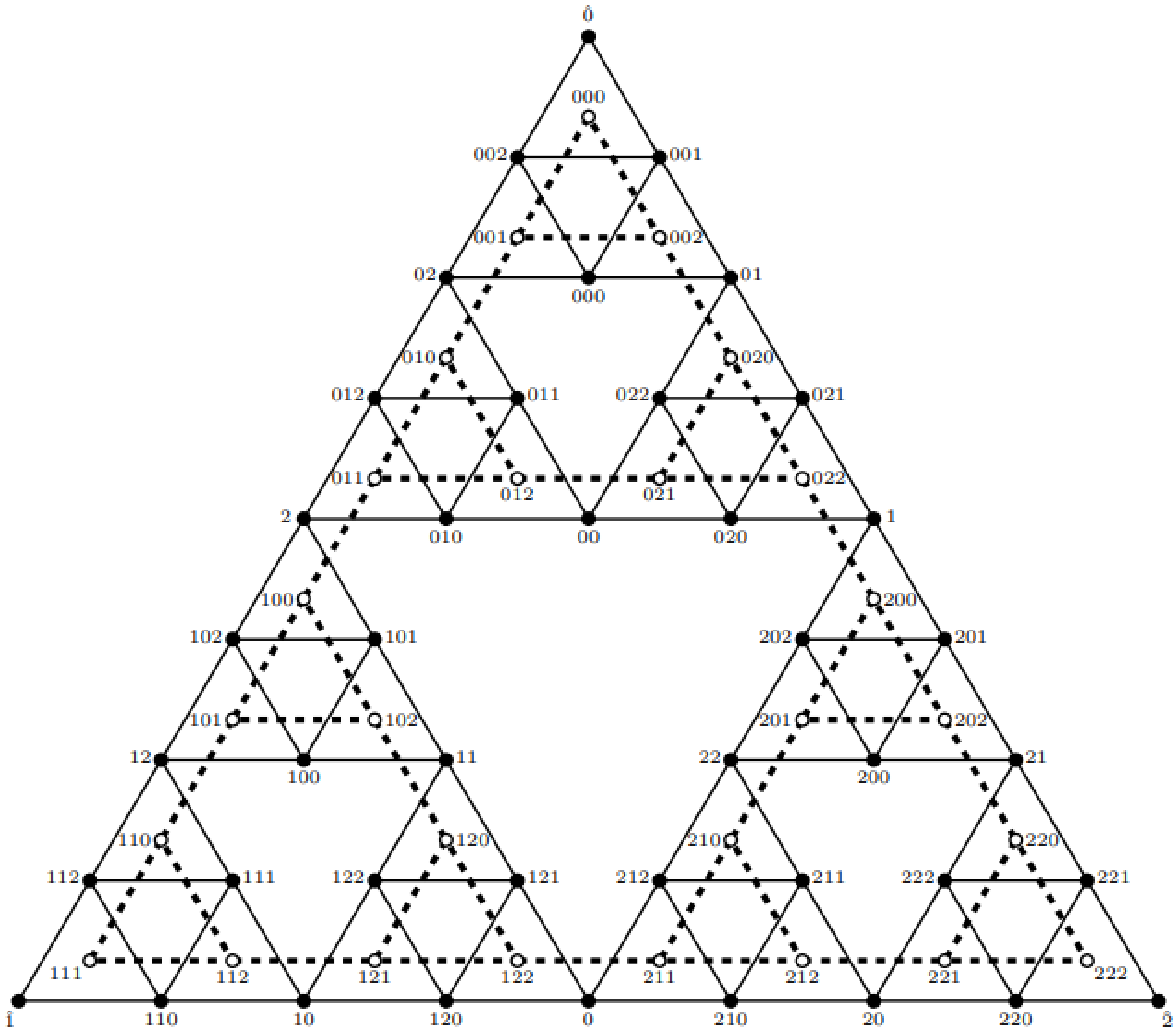

Figure 4: Sierpiński traingle graph $\hat{S}_3^3$ with Sierpiński graph S_3^3 embedded within

1. The edges with a primitive end vertex is represented as $\{\hat{k}, k^n\hat{jk}\}$. Such edges are p(p-1) in number.

2. The second type of edges are of form $\{\underline{s}\hat{ij}, \underline{s}\hat{ik}\}$. Such edges are $\frac{1}{2}(p-1)(p-2)p^{n+1}$.

3. The third and last type of edges are of form $\{\underline{s}ki^{n-v}\hat{ij}, \underline{s}\hat{ik}\}$. Such edges are p(p-1)($p^n$-1) in number.

The primitive vertices of $\hat{S}_p^n$ have degree p-1, same as the extreme vertices of S_p^{n+1}. However, all other vertices of $\hat{S}_p^n$ have degree 2(p-1), as these vertices are formed by contracting an edge. In S_p^{n+1}, all but the extreme vertices have degree p. Let, $x, y \in V(S_p^{n+1}) - \{k | k \in P\}$. Then deg(x)=deg(y)=p. On contracting edge xy, the two vertices merge into one and each loses an edge from the total edges incident on them. Thus the new vertex xy has degree 2(p-1).

10 Metric Properties of $\hat{S}_3^n$

Theorem 10.0.1 (Proposition 1 in [1])**.** *For $n \in \mathbb{N}$ and $v \in [n]_0$, and all $s, t \in V(\hat{S}^v)$ we have*
$d^{(n)}(s,t) = 2^{n-v} d^{(v)}(s,t)$
The number of shortest s,t-paths in $\hat{S}^n$ is same as in $\hat{S}^v$.
(The canonical distance function on $\hat{S}^n$ defined by $d^{(n)}$ is comprised of the sums of the lengths of the shortest paths.

Proof. **Base Case:** The theorem holds for n=0.
Induction Hypothesis: Let us assume that the theorem holds for $\hat{S}_3^n$.
Inductive Step: $\hat{S}_3^{n+1}$ is obtained from $\hat{S}_3^n$ by replacing each edge with a path of length 2. The extra vertex lies in T^{n+1} and produces two new incident edges with their other endpoints in T^{n+1} as well. The new edges do not belong to a shortest s,t-path. So, every shortest s,t-path in $\hat{S}_3^{n+1}$ stems from precisely one shortest s,t-path in $\hat{S}_3^n$ whose length has doubled. □

Theorem 10.0.2 (Theorem 1 in [1])**.** *For $k, l \in T$, there is a unique shortest $\hat{k}, \hat{l}$-path in $\hat{S}^0$ with length*
$d^{(0)}(\hat{k}, \hat{l}) = (k \neq l)$
$[(k \neq l) = 1$ *if* $k \neq l$ *and 0 if* $k = l]$
For $v \in \mathbb{N}$ and $s \in T^v$ we have:
$d^{(v)}(s, \hat{l}) = 1 + (s_1 = l) + \sum_{d=2}^{v} (s_d \neq l) 2^{d-1}$
and there are 1+$(s_1 = l)$ shortest s, $\hat{l}$-paths in $\hat{S}^v$.

Proof. If $s = s_{v+1}\overline{s} \in T^{v+1}$,
$d^{(v+1)}(s, \hat{l}) = d^{(v)}(s, \hat{l}) + (s_{v+1} \neq l) 2^v$
$=1+(s_1 = l) + \sum_{d=2}^{v} (s_d \neq l) 2^{d-1} + (s_{v+1} \neq l) 2^v$
$=1+(s_1 = l) + \sum_{d=2}^{v+1} (s_d \neq l) 2^{d-1}$ □

For S_3^{n+1}, the distance between vertices $i\overline{s}$ and j is represented by $d^{n+1}(i\overline{s}, j)$. There are two possible cases[1]:

- For $i \neq j$, any shortest path from $i\overline{s}$ to j will correspond to a shortest path in $\hat{S}^n$ from $\overline{s}$ to $\hat{\text{k}}$, with k=3-i-j and i is concatenated to the left of the vertices on that path and the last vertex being replaced by j.
- For $i = j$, the shortest path may run through either of the vertices $k \in T$, $k \neq i$, and we have to choose the shorter one or both if they are equal in length, that is, $d^{(n+1)}(i\overline{s}, i) = min\{d^{(n)}(\overline{s}, \hat{k}) | k \in T - \{i\}\} + 2^n$.

If $\overline{s} = i^n$, then by theorem 4.0.2
$d^{(n)}(\overline{s}, \hat{k}) = 2^n - 1$
If $\overline{s} = i^m \underline{s}$ with maximal $k \in [n]_0$
$d^{(n)}(\overline{s}, \hat{k}) = 1 + (s_1 = k) + \sum_{d=2}^{n-m} (s_d \neq k) 2^{d-1} + \sum_{d=n-m+1}^{n} (i \neq k) 2^{d-1}$
When m=n-1, $d^{(n+1)}(i\overline{s}, j)$ is minimal for $k \neq s_1$, that is, for both $k \in T - i$ if $s_1 = i$ and otherwise exclusively for $k = 3 - i - s_1$.

11 Metric Properties of Sierpiński Triangle Graphs ($\hat{S}_p^n$)

First we state some preliminary metric properties of $\hat{S}_p^n$.

- For $s, t \in V(\hat{S}_p^n)$ and $v \in [n]_0$,
  $d^{(n)}(s,t) = 2^{n-v} d^{(v)}(s,t)$[1]

- $d^{(0)}(\hat{k}, \hat{l}) = (k \neq l)$[1]

- $d^{(v)}(\underline{s}i\hat{j}, \hat{l})$==$1+(i \neq j \neq l) + \sum_{d=2}^{v}(s_d \neq l)2^{d-1}$.[1]
  There are 1+ $(i \neq j \neq l)$ shortest paths.

- diam($\hat{S}_p^n$)=2^n[1]

Now we move on to the distances of the various types of vertices.

11.1 Distance from Primitive Vertices

In this subsection, we discuss the properties related to the distance of the primitive vertices.

- For $p \geq 3$ and $\forall s \in V(\hat{S}_p^n)$,
  $\sum_{l=0}^{p-1} d^{(n)}(s, \hat{l}) = (p-1)2^n$[1]

- Total distance of a given primitive vertex, say $\hat{l}$
  $d^{(n)}(\hat{l}) = \frac{|\hat{S}_p^n| \sum_{l=0}^{p-1} d^{(n)}(s,\hat{l})}{p}$
  $= \frac{p}{2}(p^n+1)(p-1)2^n \frac{1}{p}$
  $= \frac{p-1}{2} 2^n (p^n+1)$[1]

Definition : For connected graph G and a fixed vertex $v \in V(G)$, we define the periphery of v as[1]:
$E_G(v) = \{w \in V(G) | d(v,w) = \epsilon(v)\}$

Theorem 11.1.1 (Theorem 2 in [1]). *For $p \in \mathbb{N}_2$ and $n \in \mathbb{N}_0$,*
$|E_{\hat{S}_{p+1}^n}(p^n)| = |V(\hat{S}_p^n)|$
Moreover, the graph induced by $E_{\hat{S}_{p+1}^n}(p^n)$ in $\hat{S}_{p+1}^n$ is equal to $\hat{S}_p^n$.

Proof. $\epsilon(\hat{p}) = diam(\hat{S}_{p+1}^n) = 2^n$
$\underline{s}i\hat{j} \in E_{\hat{S}_{p+1}^n}(p^n) \iff |\{i,j,p\}| = 3$ and $\forall d \in [v-1] : s_{1+d} \neq p$
$\iff \{i,j\} \in^P C_2$ and $\forall d \in [v-1] : s_{1+d} \in P$
$\iff \underline{s}i\hat{j} \in V(\hat{S}_p^n)$ □

11.2 Distance from Critical Vertices

For the distance between non-primitive vertices, the task may be reduced if the vertices have a common prefix $\underline{r} \in P^m$, where $m \in \mathbb{N}_0$. If $v \in \mathbb{N}$, $\mu \in [v]$, $\underline{s} \in P^{v-1}$, $t \in P^{\mu-1}$, and $i, j, k, l \in^P C_2$, then[1]:
$d^{(m+v)}(\underline{r}\underline{s}i\hat{j}, \underline{r}\underline{t}k\hat{l}) = d^{(v)}(\underline{s}i\hat{j}, \underline{t}k\hat{l}) = 2^{-m} d^{(m+v)}(\underline{s}i\hat{j}, \underline{t}k\hat{l})$

Each shortest path between two generic vertices must go through either one or two critical vertices. If the shortest path passes through three generic vertices, then the path is from some critical vertex $\hat{ij}$ to some other critical vertex $\hat{kl}$ via another critical vertex halfway through the path, because the passage through two subgraphs isomorphic to $\hat{S}_p^n$ already contribute to $2 \cdot 2^n$ to the length of the path, thus exceeding the diameter of $\hat{S}_p^{n+1}$ in all other cases.
The shortest path between $\underline{rs}\hat{ij}$ and $\underline{rt}\hat{kl}$ has a unique shortest path if and only if $|\{i,j,k,l\}| < 4$ and for $|\{i,j,k,l\}| = 4$, we have as many as four shortest paths, namely those passing through critical vertices $\hat{ik}$, $\hat{il}$, $\hat{jk}$, or $\hat{jl}$.
Considering the generic vertex $is = i\underline{s}\hat{jk}$, we want to find the shortest path from is to $\hat{lm}$. There are two possible scenarios for the shortest path from is to $\hat{lm}$[1]:

- (Case 1: $|\{i,l,m\}| = 2$) For i=l, $d^{(n+1)}(i\underline{s}\hat{jk}, \hat{im}) = d^{(n)}(\underline{s}\hat{jk}, \hat{m})$ with $1+(|\{j,k,m\}| = 3)$ shortest paths.

- (Case 2: $|\{i,l,m\}| = 3$) In this case we need to compare two distances and decide which is the shortest path.
$d^{(n+1)}(i\underline{s}\hat{jk}, \hat{lm}) = min\{d^{(n+1)}(i\underline{s}\hat{jk}, \hat{il}) + d^{(n+1)}(\hat{il}, \hat{lm}), d^{(n+1)}(i\underline{s}\hat{jk}, \hat{im}) + d^{(n+1)}(\hat{im}, \hat{lm})\} = min\{d^{(n+1)}(\underline{s}\hat{jk}, \hat{l}), d^{(n+1)}(\underline{s}\hat{jk}, \hat{m})\} + 2^n$
Comparing $d^{(n)}(\underline{s}\hat{jk}, \hat{l})$ and $d^{(n)}(\underline{s}\hat{jk}, \hat{m})$ is equivalent to comparing ρ_n with 0, where $\rho_v = \sum_{d=1}^{v} \beta_d \cdot 2^{d-1}$, $v \in [n+1]_0$.
Here, $\beta_1 = (j \neq l)(k \neq l) - (j \neq m)(k \neq m)$ and $\beta_d = (s_d = m) - (s_d = l)$ for $d \in [n] - 1$. $[\beta_d \in \{-1, 0, 1\}, |\rho_v| \leq 2^v - 1]$.
Now, the comparison of ρ with 0 is equivalent to the comparison of ρ_{n-1} with $-\beta_n \cdot 2^{n-1}$. Here, if $\beta_n = 1$ then $\rho \geq 0$ and if $\beta_n = -1$ then $\rho \leq 0$. Otherwise we compare ρ_{n-1} with 0 and we repeat the process by replacing $v = n$ with $v - 1$ and we use the new input s_v as long as $v \in \mathbb{N}$. We continue until we get $\rho \geq 0$, $\rho \leq 0$, or until we reach β_1 and if $\beta_1 = 0$ then $\rho == 0$.

s_d	β_d	sign	shortest path via	number of shortest paths
m	1	>	$\hat{im}$	$1+(\|\{j,k,m\}\| = 3)$
$\alpha\hat{m}$	1	>	$\hat{im}$	1
α	0	undecided		
$\hat{lm}$	0	=	$\hat{il}$ or $\hat{im}$	2
$\alpha_1\hat{\alpha}_2$	0	=	$\hat{il}$ or $\hat{im}$	4
l	-1	<	$\hat{il}$	$1+(\|\{j,k,l\}\| = 3)$
$\hat{\alpha l}$	-1	<	$\hat{il}$	1

Table 1: Decision table for $d^{(n+1)}(i\underline{s}\hat{jk}, \hat{lm}), |\{i,l,m\}| = 3; \alpha_{(1,2)} \in P - \{l,m\}, d \in [n]$ [Table 1 in [1]]

An algorithm that decides about the shortest path to a fixed critical vertex $\hat{lm}$ can be based on the simple finite automaton of figure 5. It consists of a transient state 0 and two absorbing states L and M. The data $s_d \in s$ for $d \in [n]$ are inserted, starting with d=n in state 0. If s_d is any of the labels of the connecting lines we move to the corresponding absorbing state and stop. If we reach absorbing state L or M, then it means that, $\hat{il}$ or $\hat{im}$

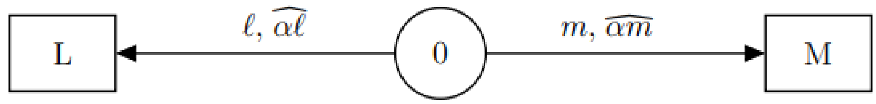

Figure 5: Decision automaton for the goal vertex $l\hat{m}$

respectively are the shortest path. For any other input we stay at state 0 and continue. If even after reaching s_1 we still stay at state 0 we infer that the two paths are equal in length.

11.3 Distance from Generic Vertices

In this subsection, we consider the distance between two generic vertices is and jt in $\hat{S}_p^{n+1}$. The shortest is,jt-path could either go directly or through any of the other subgraphs isomorpic to $\hat{S}_p^n$.
The distance corresponding to the direct path is denoted by[1]:
$d_p^{(n+1)}(is, jt) = d^{(n+1)}(is, i\hat{j}) + 2^{n-v}d^{(v+1)}(jt, \hat{ij})$
$=d^{(n)}(s, \hat{j}) + 2^{n-v}d^{(v)}(t, \hat{i})$
We denote the distance corresponding to the shortest path through any of the other subgraphs isomorphic to $\hat{S}_p^n$ by[1]:
$d_k^{(n+1)}(is, jt) = d^{(n+1)}(is, i\hat{k}) + d(i\hat{k}, j\hat{k}) + 2^{n-v}d^{(v+1)}(jt, j\hat{k})$
$=d^{(n)}(s, \hat{k}) + 2^n + 2^{(n-v)}d^{(v)}(t, \hat{k})$, where $k \in [p]_0$
Next we compare the two types of possible paths and decide which is the shorter one, thus achieving the shortest is,jt-path. Thus we write the shortest is,jt-path as[1]:
$d^{(n+1)}(is, jt) = min\{d_k^{(n+1)}(is, jt) | k \in [p+1]_0 - \{i, j\}\}$.

To decide for which k the minimum is attained, we compare for $k \in P - \{i, j\}$. We can observe that the comparison between $d_k^{(n+1)}(is, jt)$ and $d_p^{(n+1)}(is, jt)$ is equivalent to the comparison of ρ_n with 2^n. For $m \in [n+1]_0$, $\rho_m = \sum_{d=1}^{m}(\sigma_d, \tau_d) \cdot 2^{d-1}$

Here, taking $s = s_n ... s_2\hat{x}y, t = t_v ... t_2\hat{w}z$ we can define[1]:

- $\sigma_d = (s_d = k) - (s_d = j)$, for $d \in [n] - 1$
- $\sigma_1 = (x \neq j)(y \neq j) - (x \neq k)(y \neq k)$
- $\tau_d = (t_{d-n+v} = k) - (t_{d-n+v} = i)$, for $d \in [n] - [n - v + 1]$
- $\tau_{n-v+1} = (w \neq i)(z \neq i) - (w \neq k)(z \neq k)$
- $\tau_d = 0$, for $d \in [n - v]$

Adding to this we know from the definitions that, $(\sigma_d + \tau_d) \in \{-2, -1, 0, 1, 2\}$ and $2 - 2^{m+1} \leq \rho_m \leq 2^{m+1} - 2$

We start in state A of the automaton beginning at d=n and enter the data $\sigma_d + \tau_d$. For $d \in [n - v]_0$, we get $\tau_d = 0$. If we end in states A or D, it means that the direct path is shorter than any bypass through a subgraph $k\hat{S}_p^n$. Ending in states C or E implies that

s_d	σ_d	t_δ	$\tau_\delta + n - v$
k or $\hat{\alpha k}$	1	k or $\hat{\beta k}$	1
α or $\hat{jk}$ or $\alpha_1\hat{\alpha_2}$	0	β or $\hat{ik}$ or $\beta_1\hat{\beta_2}$	0
j or $\hat{\alpha j}$	-1	i or $\hat{\beta i}$	-1

Table 2: Decision table for $d^{(n+1)}(is, jt); \alpha_{(1,2)} \in P - \{j, k\}, \beta_{(1,2)} \in P - \{i, k\}, d \in [n], \delta \in [v](\tau_d = 0 \text{ for } d \in [n - v])$[Table 2 in [1]]

the bypass is necessary and it can only be reached if one of the inputs is $\sigma_d + \tau_d = 2$. However, if we end in state B, then we can infer that both paths are equal. The decision automaton has been represented in Figure 3. (All states A, B, C, D, and E refer to the states in Romik's automaton.)[1][3]

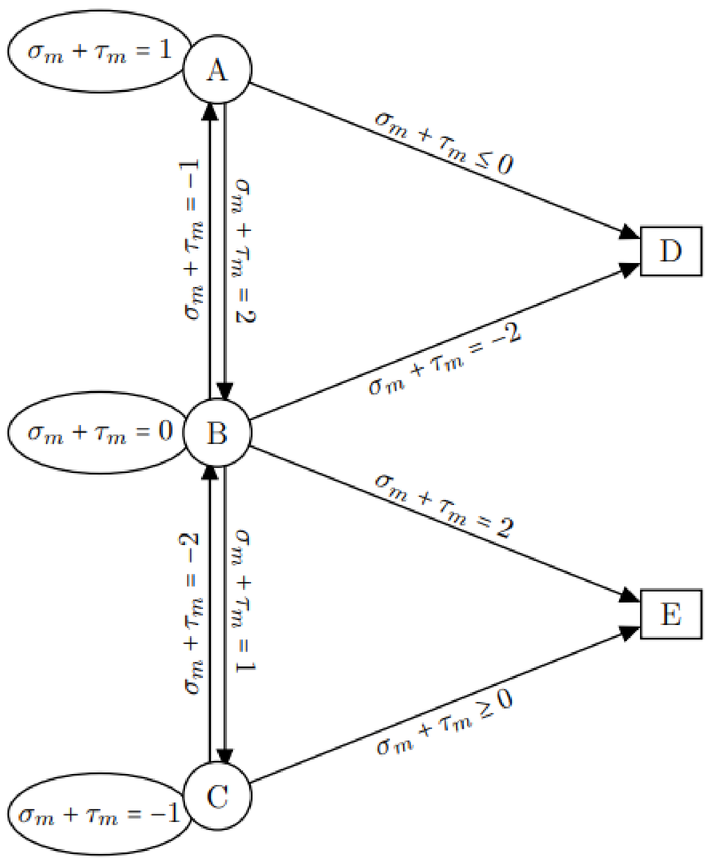

Figure 6: Decision automaton for shortest path for generic vertices

For the Sierpiński triangle graphs, we have to combine all admissible shortest paths to and from the intermediate critical vertices. There are at most 8 optimal paths between any 2 vertices.

11.4 Additional Metric Properties

We state some additional metric properties of $\hat{S}_p^n$

Definition : $x_p(\lambda) = |\{s \in V(\hat{S}_p^n) | d^{(n)}(s, \hat{l}) = \lambda\}|$ for $\lambda \in [2^n]$ and fixed $l \in P$[1]

Theorem 11.4.1 (Proposition 2 in [1]). *For $n \in \mathbb{N}$ and $\{l, m\} \in^T C_2$,*
$d^{(n)}(\hat{lm}) = \frac{7}{10}6^n + 2^n - \frac{1}{5}$
in $\hat{S}_3^n$, that is, asymptotically, for large n, the average distance to a fixed critical vertex is seventy-percent of the average distance to a fixed primitive vertex.

Proof. We may assume , l, m=1, 2 Let, $C_n = \{0^v\hat{12} | v \in [n]_0\} \bigcup \{\hat{0}\}$, that is the set of the vertices in the central vertical axis of $\hat{S}_3^n$, and $|C_n| = n + 1$.
Now, we define $c_n = \sum_{c \in C_n} d^{(n)}(c, \hat{1})$ and we get a recurrence relation:
$c_0 = 1$, and $\forall n \in \mathbb{N}_0$ we get, $c_{n+1} = c_n + (n + 2) \cdot 2^n$.
The solution to this recurrence relation can be expressed as $c_n = n \cdot 2^n + 1$.
Let, S_n denote the set of those vertices of $\hat{S}_3^n$ which lie strictly to the left of the central vertical axis. Therefore, $|S_n| = \frac{1}{2}(|\hat{S}_3^n| - |C_n|) = \frac{1}{4}(3^{n+1} - 2n + 1)$
For $s_n = \sum_{s \in S_n} d^{(n)}(s, \hat{1})$, we get the recurrence:
$s_n = 0$, and $\forall n \in \mathbb{N}_0$ we get, $s_{n+1} = d^{(n)}(\hat{1}) - 2^{n+1} + s_n + |S_n| \cdot 2^n = s_n + (7 \cdot 3^n - 2n - 3) \cdot 2^{n-2}$.
This recurrence relation has the solution $s_n = \frac{7}{20}6^n - \frac{2n-1}{4}2^n - \frac{3}{5}$
$\therefore d^{(n+1)}(\hat{12}) = 2d^{(n)}(\hat{1}) + 2s_n + c_n + \frac{1}{2}(3^{n+1} - 1) \cdot 2^n$
$\implies d^{(n)}(\hat{12}) = \frac{7}{10}6^n + 2^n - \frac{1}{5}$ □

Theorem 11.4.2 (Proposition 3 in [1]). *Let, $p \in \mathbb{N}_2$ and $n \in \mathbb{N}_0$. Then $\forall v \in [n]_0$, and $\forall m \in [2^{n-1-v}]_0$*
$x_p((2m + 1)2^v) = \frac{1}{2}(p - 1)^{q(m)+1}((p - 1)^v + 1)$ *and* $x_p(2^n) = \frac{p-1}{2}((p - 1)^n + 1)$
(q(m)= number of non-zero bits in binary representation of m)

Proof. We will use induction to prove this theorem. Base Case: For m=0, v=0, we get $x_p(1) = p - 1$
Induction Hypothesis: Let us assume that the theorem holds true for uptill $x_p(2^n)$.
Inductive Step: Every $\lambda \in [2^{n+1} - 1]$ can be uniquely represented in binary with $v \in [n+1]_0$: $\lambda = (\lambda_n ... \lambda_{v+1} 10^v)_2$
$\lambda = (2m + 1) \cdot 2^v$ with m=$(\lambda_n ... \lambda_{v+1})_2 \in [2^{n-v}]_0$
From the recursive definition of Sierpiński triangle graphs, $\lambda_n = 0$, that is $\lambda \in [2^n - 1]$, then x_p is the same as in S_p^n and also the m is the same. Also for $\lambda = 2^n$, that is, v=n and m=0, $x_p(2^n) = \frac{p-1}{2}((p - 10^n + 1)$
For $\lambda \in [2^{n+1} - 1] - [2^n]$,
$x_p(\lambda) = (p - 1)x_p(\lambda - 2^n)$
$\therefore$ The theorem holds for $\lambda \in [2^{n+1} - 1]$
Let $\lambda \in 2^{n+1}$, and we subtract the number of intersection points of the p-1 copies to avoid double counting while calculating $x_p(2^{n+1})$
$\therefore x_p(2^{n+1}) = (p - 1)x_p \cdot 2^n -^{p-1} C_2$
$\implies x_p(2^{n+1}) = \frac{p-1}{2}((p - 1)^{n+1} + 1)$
Thus, the theorem holds for all values of $m, v \in \mathbb{N}_0$ □

Theorem 11.4.3 (Proposition 4 in [1]). *Let, $p \in \mathbb{N}_2$, $i \in P$, and $n \in \mathbb{N}_0$. Then $\forall s \in V(i\hat{S}_p^n)$:*
$\epsilon(s) = max\{d^{(n+1)}(s, \hat{l}) | l \in P - \{i\}\}$

Proof. Let, $p \in \mathbb{N}_3$. This assumption ensures that $\epsilon(s) > 2^n$
Let, $t \in V(\hat{S}_p^{n+1})$ be eccentric with respect to s, that is, $d^{(n+1)}(s,t) = \epsilon(s)$
Then $t \in V(j\hat{S}_p^n)$ for some $j \in P - i$ because otherwise $d^{(n+1)}(s,t) \leq 2^n$
$\therefore d^{(n+1)}(s,t) \leq d^{(n+1)}(s,\hat{ij}) + d^{(n+1)}(\hat{ij},t) \leq d^{(n+1)}(s,\hat{ij}) + 2^n = d^{(n+1)}(s,\hat{ij}) + d^{(n+1)}(\hat{ij},\hat{j}) = d^{(n+1)}(s,\hat{j})$
$\therefore \hat{j}$ is also eccentric with respect to s as well. □

Theorem 11.4.4 (Theorem 3 in [1]). *For $p \in \mathbb{N}_2$ and $n \in \mathbb{N}_0$, the periphery of $\hat{S}_p^n$ is given by:*
$P(\hat{S}_p^n) = \hat{P} \bigcup \{s_v...s_2\hat{ij} \in V(\hat{S}_p^n) | \{i,j,s_2,...,s_v\} \neq P\}$
For $n \in [p-1]_0$, we have $rad(\hat{S}_p^n) = 2^n$ and $C(\hat{S}_p^n) = V(\hat{S}_p^n)$ is the center of $\hat{S}_p^n$.
If $n \in \mathbb{N}_{p-1}$, then $rad(\hat{S}_p^n) = 2^{n-p+1} \cdot (2^{d-1} - 1)$ and $C(\hat{S}_p^n) = \{s_{p-1}...s_2\hat{ij} \in V(\hat{S}_p^n) | \{i,j,s_2,...,s_{p-1}\} = P\}$, such that $|C(\hat{S}_p^n)| = \frac{1}{2}p!$

12 Conclusion

In this report we tried to look at the Tower of Hanoi game and one of it's many variations, the Switching Tower of Hanoi game. We also took a look at the Hanoi graph and Sierpiński graph and the various properties related to them. Lastly, we looked into the properties of a variation of the Sierpiński graphs, that is, the Sierpiński Triangle graphs and their properties.

References

[1] Andreas M. Hinz, Caroline Holz auf der Heide, Sara Sabrina Zemljič, *Metric Properties of Sierpiński Triangle Graphs.*, 2022.

[2] Andreas M. Hinz, Sandi Klavžar, Sara Sabrina Zemljič, *A survey and classification of Sierpiński-type Graphs.*, 2016.

[3] Andreas M. Hinz, Caroline Holz auf der Heide, *An efficient algorithm to determine all shortest paths in Sierpiński Graphs.*, 2014.

[4] A.M. Hinz, S. Klavžar, C. Petr, *The Tower of Hanoi- Myths and Maths.*, Second Ed., Springer, Cham, 2018.

[5] John M. Harris, Jeffry L. Hirst, Michael J. Mossinghoff, *Combinatorics and Graph Theory.*, Springer, 2000.

[6] Kannan Balakrishnan, Manoj Changat, Andreas M. Hinz, Divya Sindhu Lekha, *The median of Sierpiński graphs.*, 2022.

[7] Martin Gardner, *Hexaflexagons, Probability Paradoxes, and the Tower of Hanoi.*, 2008.