## Linnæus University <br> \author{ Sweden 

}
## Degree project

## The Topswop Forest



Author: Desheng Zhang
Supervisor: Jonas Nordqvist Examiner: Hans Frisk
Date: 2021/6/10
Course Code: 2MA41E
Subject: Mathematics
Level: Bachelor

# The Topswop Forest 

Desheng Zhang

June 10, 2021


#### Abstract

In this thesis, we will define the topswop forest and study the properties of the forest. We will show the number of trees and leaves in the forest. We will also do an experiment to show there is more than an exponential growth between the number of nodes of each tree and the number of elements in the permutation. The experiment also shows that the tallest tree doesn't always contain the identity permutation. In the later section, we derive a linear lower bound for the topswop problem by studying a specific family of permutation.


## Contents

1 Introduction ..... 2
1.1 The pancake problem ..... 2
1.2 The topswops problem ..... 2
1.3 Content of this thesis ..... 3
2 Topswop Forest ..... 3
2.1 The topswop function ..... 3
2.2 Properties of the topswop forest ..... 6
2.3 The analysis of the number of nodes and flips for each topswop tree ..... 10
2.4 Bounding the topswops problem ..... 12
2.4.1 The linear lower bound ..... 12
2.4.2 The best proven quadratic lower bound ..... 16
2.4.3 The Wilf upper bound ..... 17
2.4.4 The Fibonacci upper bound ..... 17
3 Discussion ..... 18
Appendices ..... 21
A Tables for the number of nodes and the largest number of flips ..... 21
B Regression results ..... 24

C Codes: Compute number of nodes for each tree
D Codes: Compute number of flips for the specific tree
E Codes: Compute the maximum number of flips in the topswop forest 26

## 1 Introduction

In discrete mathematics, a permutation is an ordered arrangement of some elements of a set. The prefix reversal is a way of rearranging the permutation by reversing the order of the first $n$ elements of the permutation, where $n$ is arbitrary. The pancake problem also known as the problem of Sorting By Prefix Reversals asks for the minimum number of prefix reversals required to sort a given permutation. The deterministic pancake problem also called the topswop problem is a variation of the pancake problem and it deals with finding the maximum number of prefix reversals, where the size of each prefix reversal is the first element of the permutation. A recent paper [5] has shown the progress of finding the maximum number of prefix reversals. The topswop problem gives rise to a dynamical system which can be described in terms of a graph. We can regard the topswop problem as a forest which is a disconnected and acyclic graph. We are interested in certain properties of this forest. Typically, one studies the height of the tallest tree, we are also interested in size of the trees and various other properties. We will start the thesis by first introducing the pancake problem.

### 1.1 The pancake problem

The pancake problem was first posed in 1975 [7]. Given a stack of $n$ pancakes in arbitrary order, all of different sizes, the aim is to sort them in as few operations as possible to obtain a stack of pancakes with sizes increasing from top to bottom. The only allowed sorting operation is a spatula flip, in which a spatula is inserted beneath any pancake, and all pancakes above the spatula are lifted and replaced in reverse order. We can regard the stack as a permutation and a flip as a prefix reversal of the permutation [3].

Example 1. We have a stack of pancakes in arbitrary order $(4,2,3,1,5)$. We choose to flip the first three pancakes and it becomes $(3,2,4,1,5)$.

### 1.2 The topswops problem

A variation on the original pancake problem is the deterministic pancake problem, also known as topswops problem [1], was first proposed by the British mathematician John Conway as one of a series of card games [9]:

A deck of cards is numbered 1 to $n$ in random order. Perform the following operations on the deck. Whatever the number on the top card is, count down that many in the deck and turn the whole block over on top of the remaining cards. Then, whatever the number of the (new) top card, count down that many cards in the deck and turn this whole block over on top of the remaining cards. Repeat the process. Show that the number 1 will eventually reach the top.

We can view the deck of cards as a permutation on $\{1,2,3, \ldots, n\}$. Suppose the first card from the deck is $k$, we can describe the topswops problem by the following algorithm:

1. Find the first card $k$ from the deck
2. Take the first $k$ cards from the deck
3. Swap these cards and place them back on the deck
4. Repeat step 1, 2 and 3 until the first card is 1.

The question follows: What is the maximum number of steps to the termination? A recent paper [5] shows the maximum number of steps for $n=18$ and $n=19$. Thus, the numerical results are known for $n \leq 19$ [10] [5], see Table 1 for their list.

| Deck Length (n) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Maximum iterations | 0 | 1 | 2 | 4 | 7 | 10 | 16 | 22 | 30 | 38 |
| Deck Length (n) | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  |
| Maximum iterations | 51 | 63 | 80 | 101 | 112 | 130 | 159 | 191 | 221 |  |

Table 1: The maximum number of iterations for each $n$
However, when $n \geq 20$ the solutions are unknown. We only have bounds for the solutions. One of the quadratic lower bound was proven by Morales and Sudborough [8] and one of the Fibonacci upper bound proven by Klamkin [6].

### 1.3 Content of this thesis

Follow the topswops problem proposed by Conway, we built a topswop forest based on the Graph Theory [2]. In Section 2.1, we define the topswop forest. In Section 2.2, we will study the number of topswop trees and leaves in the forest. Section 2.3 will go through the results of the experiments for the sizes of trees. A lower bound for the size of trees will also be contained in the end of this section. In Section 2.4, we will focus on finding the tallest tree in the forest. In other words, we are going to study on the bounds of the topswops problem. We will develop a linear lower bound by finding the number of flips of a specific family of permutation. We will also present the proven bounds in the end of the section.

## 2 Topswop Forest

In this section, we are going to study the topswop forest. We will first introduce the topswop function and define the topswop forest. Then, we will study some properties of the topswop forest.

### 2.1 The topswop function

Let $S_{n}$ denote the symmetric group of permutations on $\{1,2, \ldots, n\}$. Let $P$ be a permutation in $S_{n}$ and we denote $P[a]$ as the $a$ th element of $P$. Now, we let $P^{\prime}$ be an element in $S_{n}$, if $P[1]=b$, we have

$$
P^{\prime}[a]= \begin{cases}P[b-a+1], & \text { if } 1 \leq a \leq b,  \tag{2.1}\\ P[a], & \text { if } a \geq b+1 .\end{cases}
$$

Then we define the topswop function $f: S_{n} \rightarrow S_{n}$ by $f(P)=P^{\prime}$.
Let $P_{i}$ denote the $i$ th iteration of the topswop function $f$. Given a permutation $P_{1}$, the iterates of $f$ will be: $f\left(P_{1}\right)=P_{2}, f\left(f\left(P_{1}\right)\right)=P_{3}, \ldots$, and the iterates will finally terminate. Now, we let run $(P)$ denote the sequence of iterates of $f$ and $|\operatorname{run}(P)|$ denote the length of this sequence. We will illustrate this concept by means of an example.

Example 2. Here we list an example of the topswop function: given a permutation $P_{1}=(4,3,1,2)$, then $P_{2}=f\left(P_{1}\right)=(2,1,3,4), P_{3}=f\left(f\left(P_{1}\right)\right)=(1,2,3,4)$ and $\left|\operatorname{run}\left(P_{1}\right)\right|=2$.

In order to continue our discussion, we need the following lemmas.
Lemma 1. For any given permutations $P_{1}$, the iterates of $f$ :

$$
P_{1} \rightarrow P_{2} \rightarrow P_{3} \rightarrow \ldots \rightarrow P_{i} \neq P_{1},
$$

where $i \geq 2$.
Proof. We prove this by contradiction. We assume by contradiction that $P_{1} \rightarrow P_{2} \rightarrow$ $P_{3} \rightarrow \ldots \rightarrow P_{i}=P_{1}$. Let $P_{j}[1]$ denote the first element of permutation $P_{j}$, where $1 \leq j \leq i$. We denote $t=\max \left\{P_{1}[1], P_{2}[1], \ldots, P_{i}[1]\right\}$. Suppose at the $r$ th iteration we have $P_{r+1}[1]=t$, at the $(r+1)$ th iteration, we have $P_{r+2}[t]=t$. The remaining iterations will always retain: $P_{r+k}[t]=t$, where $k \geq 2$.

Now, suppose if $0 \leq r \leq i-1$. Then we have: $P_{i}[t]=t, P_{1}[t]=t, P_{2}[t]=t, \ldots, P_{i}[t]=$ $t$. This means $t$ will not appear on the first element anymore during the iterations and we get: $t \neq \max \left\{P_{1}[1], P_{2}[1], \ldots, P_{i}[1]\right\}$ which leads to a contradiction. Thus we prove the lemma.

Lemma 1 shows that there are no repeated sequences during the iterates of the topswop function.

Now, suppose $P_{i}$ and $P_{i+1}$ are the elements in $S_{n}$, where $P_{i+1}=f\left(P_{i}\right)$. We can see that $P_{i}$ is a preimage of $P_{i+1}$. We say a permutation $P_{i} \in S_{n}$ can be traversed back if it has a preimage.

Lemma 2. A permutation $P_{i} \in S_{n}$ can be traversed back if and only if $P_{i}[a]=a$ for some $a$, where $1 \leq a \leq n$.

Proof. We assume $P_{i}$ can be traversed back. Then $P_{i}$ has a preimage $P_{i-1}$, we denote $P_{i-1}[1]=a$. By formula (2.1) we can derive:

$$
P_{i}[a]=P_{i-1}[a-a+1]=P_{i-1}[1]=a
$$

Now we assume $P_{i}[a]=a$. Then we can derive the preimage:

$$
P_{i-1}[b]= \begin{cases}P_{i}[a-b+1], & \text { if } 1 \leq b \leq a  \tag{2.2}\\ P_{i}[b], & \text { if } b \geq a+1\end{cases}
$$

We will also introduce the concept of Wilf Number which was proposed by Wilf [4]. Given a permutation $P$, there are $m$ numbers in their original positions: $P\left[a_{j}\right]=a_{j}$, where $1 \leq j \leq m$. We can make the following definition.

Definition 2.1 (Wilf number). We denote $w$ as the Wilf number, then we have

$$
w=2^{\left(a_{1}-1\right)}+2^{\left(a_{2}-1\right)}+\ldots+2^{\left(a_{j}-1\right)}+\ldots+2^{\left(a_{m}-1\right)}=\sum_{j=1}^{m} 2^{\left(a_{j}-1\right)}
$$

Example 3. Here, we give some examples of Lemma 2 and the Wilf number:

1. Given a permutation $P=(3,1,2,4,5)$, we can see that $P[4]=4$, the Wilf number $w=2^{(4-1)}=8$ and by function $(2.2)$ we can derive the preimage $(4,2,1,3,5)$.
2. Given a permutation $P=(1,4,3,2,5)$, we can see that $P[3]=3$ and $P[5]=5, w=$ $2^{(3-1)}+2^{(5-1)}=68$ and we can derive two preimages $(3,4,1,2,5)$ and $(5,2,3,4,1)$.
3. Given a permutation $P=(1,3,2,5,4)$, we can see that $P[1]=1, w=2^{(1-1)}=1$ and we can derive the preimage $(1,3,2,5,4)$.
4. Given a permutation $P=(3,4,5,2,1)$, we have $P[i] \neq i$ for $1 \leq i \leq 5$, the Wilf number is zero and the permutation can not be traversed back.

Case 2 illustrates that the preimage of an element in $S_{n}$ under the topswops function may contain more than one element. Case 3 shows that the preimage of an element in $S_{n}$ can be itself. Case 4 shows an element in $S_{n}$ with empty preimage.

Now, we denote $\left|S_{n}\right|$ as the total number of permutations of $S_{n}$. We get the following lemma:

Lemma 3. For any given permutation $P_{1} \in S_{n}$, the iterates of $f$ will finally terminate at the rth iteration, where $0 \leq r \leq\left|S_{n}\right|-1$.

Proof. Case 1: $P_{1}[1]=1$
The topswop algorithm will terminate directly and $r=0$.
Case 2: $P_{1}[1] \neq 1$
By Lemma 1, we know that each permutation is different during the iterates of $f$. Since the maximum number of possible permutations is $\left|S_{n}\right|$, we can get $1 \leq r \leq\left|S_{n}\right|-1$.

By the above lemma, we know that for any given permutation $P_{1} \in S_{n}$, the iterates of $f$ will finally terminate. Since the topswop algorithm terminates when the first element of permutation is 1 , we can derive the following corollary.

Corollary 3.1. The iterates of $f$ will finally terminate with $P_{1+r}[1]=1$.
Corollary 3.1 provides an answer to Conway's original problem which is mentioned in Section 1.2.

### 2.2 Properties of the topswop forest

Now, we denote $R$ as all the permutations starting with 1 , where $R \subseteq S_{n}$. Let $r_{i} \in R$, where $1 \leq i \leq(n-1)$ !. We let $D_{i} \subseteq D \subseteq S_{n}$, where $D$ represents all the derangements of $S_{n}$ and $D_{i}$ is a set of derangements that finally terminate at $r_{i}$. We denote $\left|D_{i}\right|$ as the number of elements in $D_{i}$ and let $d_{j} \in D_{i}$, where $1 \leq j \leq\left|D_{i}\right|$. Let $T_{i}$ be a graph where $D_{i}$ and all the permutations during iterates of $f\left(d_{j}\right), j=1,2, \ldots,\left|D_{i}\right|$ are the vertices. Let $V_{j}$ and $V_{k}$ be any two vertices in $T_{i}$. Then we connect $V_{j}$ and $V_{k}$ by an edge which directed from $V_{j}$ to $V_{k}$ if $V_{j}$ is a preimage of $V_{k}$ under the function of $f$.

Lemma 4. $T_{i}$ is a rooted tree.
Proof. By Lemma 2 and 3 we know that $T_{i}$ is a connected graph. By Lemma 1 we know that $T_{i}$ does not have a cycle. By Lemma 1 we know that each node except the root has exactly one parent. By Lemma 1 and 3 we know that every edge is directed towards the root. Thus, $T_{i}$ is a rooted tree.

We can now define the topswop tree and forest:
Definition 2.2 (Topswop Tree). $T_{i}$ is a Topswoptree, where $r_{i}$ has been designated as the root and $D_{i}$ are its leaves.

Example 4. An example of a topswop tree is illustrated in Figure 1, where (1, 2, 3, 4) is the topswop root. $(2,4,1,3),(4,3,2,1),(4,3,1,2),(4,1,2,3)$ and $(3,1,4,2)$ are the topswop leaves. All the edges are directed towards the root.


Figure 1: Illustration of a topswop tree

Definition 2.3 (Topswop Forest). A topswop forest, denoted by $G_{n}$, contains all the topswop trees.

Example 5. An example of a topswop forest is illustrated in Figure 2. There are 6 independent trees in the forest.


Figure 2: Illustration of a topswop forest

We can now study on the properties of the topswop forest. By the definition of the topswop tree, we can easily derive the number of trees in the topswop forest. We can make the following proposition.

Proposition 1. The number of trees in $G_{n}$ is $(n-1)$ !.
Proof. According to the definition of topswop tree, we can count the number of trees by finding the number of topswop roots. Since the roots are the permutations starting with 1 in $S_{n}$, we can know there are in total $(n-1)$ ! number of roots.

Then, we study on the number of leaves in the topswop forest. Since a leaf is a permutation that can not be traversed back. We can regard the leaves as the derangements of such that no element appears in its original position. We can find the number of leaves in $G_{n}$ by counting the number of derangements of an $n$-set.

Proposition 2. The number of leaves in $G_{n}$ is $n!\sum_{i=0}^{n} \frac{(-1)^{i}}{i!}$.
Proof. First, we need to find all permutations in which each element appears in its original location. For $1 \leq k \leq n$ we define $S_{k}$ to be the set of permutations which leave $k$ in its natural position. By inclusion-exclusion principle we can derive:

$$
\begin{aligned}
\left|S_{1} \cup \ldots \cup S_{n}\right| & =\sum_{i}\left|S_{i}\right|-\sum_{i<j}\left|S_{i} \bigcap S_{j}\right|+\sum_{i<j<k}\left|S_{i} \bigcap S_{j} \bigcap S_{k}\right|+\ldots+(-1)^{n+1}\left|S_{1} \bigcap \ldots \bigcap S_{n}\right| \\
& =\binom{n}{1}(n-1)!-\binom{n}{2}(n-2)!+\binom{n}{3}(n-3)!-\ldots+(-1)^{n+1}\binom{n}{n} 0! \\
& =\sum_{i=1}^{n}(-1)^{i+1}\binom{n}{i}(n-i)! \\
& =\sum_{i=1}^{n}(-1)^{i+1} \frac{n!}{(n-i)!i!}(n-i)! \\
& =n!\sum_{i=1}^{n} \frac{(-1)^{i+1}}{i!}
\end{aligned}
$$

Then, we can derive the number of derangements:

$$
n!-\left|S_{1} \bigcup \ldots \bigcup S_{n}\right|=n!\sum_{i=0}^{n} \frac{(-1)^{i}}{i!}
$$

This proves the proposition.

When $n$ becomes big we can approximate the number of leaves.
Corollary 4.1. When $n$ is large we can approximate the number of leaves to $\frac{n!}{e}$.
Proof. By the power series of $e^{x}$ :

$$
e^{x}=\sum_{i=0}^{\infty} \frac{x^{i}}{i!}
$$

we can obtain using $x=-1$ :

$$
\lim _{n \rightarrow \infty} \sum_{i=0}^{n} \frac{(-1)^{i}}{i!}=\frac{1}{e}
$$

By Proposition 2, we can approximate the number of leaves to $\frac{n!}{e}$ when $n$ is large.
The above corollary illustrates that when $n$ increases the number of leaves in the forest will have a factorial growth.

### 2.3 The analysis of the number of nodes and flips for each topswop tree

An experiment has been done by running a program in Mathematica. We ran the experiment for $n \leq 11$ and derived the results for the number of nodes and the largest number of flips of each topswop tree. For simplicity, we ranked the first 6 trees with the largest number of nodes for each $n$ and listed the results in the tables listed in Appendix A. The first column shows the root of each tree. The second column shows the number of nodes and the third column shows the largest number of flips.

From the tables, we can see that the trees containing the identity permutation has the largest number of nodes. We also show the number of the nodes in the bar chart in Figure 3. For simplicity, we only pick the trees containing the family of permutations $(1,2,3,4, \ldots n),(1,3,2,4, \ldots, n),(1,4,3,2, \ldots, n),(1,3,4,2, \ldots, n)$, where $2 \leq n \leq 11$. The vertical axis shows the number of nodes in natural logarithm and the horizontal axis shows the number of elements in the permutation. The yellow line is a least squares approximation by $a+b n+c n^{2}$, the parameters $a, b$ and $c$ are given by: $a=-0.279$, $b=0.379$ and $c=0.075$. We also did a hypothesis test for the coefficient $c$ and the result shows that $c$ is significant. Thus, there is more than an exponential growth for the number of nodes against the number of elements in the permutation.


Figure 3: Number of nodes of topswop trees (log scale)

Now, we check the largest number of flips in the tables. We can see that the tree containing the identity permutation tends to have the largest number of flips. However, this is not always the case. We have the following results.

Theorem 1. The longest sequence of iterations does not always end up in the identity permutation $(1,2,3, \ldots, n)$.

Proof. We prove this by means of a counter-example. In table 6 from Appendix A, we can see that the largest numbers of flips are both 10 for the topswops tree end up in the identity permutation and in the permutation ( $1,4,3,2,5,6$ ).

Getting exact results about the sizes of the trees seem difficult. However, we may produce the lower bound on its size. We let $T_{n}$ denotes the tree in $G_{n}$, where the identity permutation $(1,2,3, \ldots, n)$ is the root and let $\left|T_{n}\right|$ represents the number of nodes of $T_{n}$.

Theorem 2. $\left|T_{n}\right| \geq 2^{n-1}$
Proof. Let $T_{n}$ denote the tree in $G_{n}$ containing $(1,2,3, \ldots, n)$.
We define $P_{i}$ as the nodes of $T_{n}$ where $1 \leq i \leq\left|T_{n}\right|$ and let $P_{1}=(1,2,3, \ldots, n)$. Then, we use the symbol $\leftarrow$ represents prefix reversal:

$$
\begin{aligned}
P_{1} & \leftarrow P_{i} \\
P_{i} & \leftarrow P_{j}
\end{aligned}
$$

where $i \neq j$ and $1 \leq i, j \leq\left|T_{n}\right|$. We append $\mathrm{n}+1$ to each node. Since the prefix reversal only apply to the first $n$ elements, we can derive:

$$
\begin{aligned}
& \left(P_{1}, n+1\right) \leftarrow\left(P_{i}, n+1\right) \\
& \left(P_{i}, n+1\right) \leftarrow\left(P_{j}, n+1\right)
\end{aligned}
$$

where $i \neq j, 1 \leq i, j \leq\left|T_{n}\right|$ and $\left(P_{1}, n+1\right)=(1,2,3, \ldots, n, n+1) \in T_{n+1}$. Thus $\left(P_{i}, n+1\right) \in T_{n+1}$.

We define $I_{i}$ as the permutation of $P_{i}$ in reverse order and we can easily derive:

$$
\left(P_{i}, n+1\right) \leftarrow\left(n+1, I_{i}\right)
$$

where $1 \leq i \leq\left|T_{n}\right|$. Thus $\left(n+1, I_{i}\right) \in T_{n+1}$.
By the previous result we can get $\left|T_{n+1}\right| \geq 2\left|T_{n}\right|$. Since $\left|T_{2}\right|=2$, we have:

$$
\begin{gathered}
\left|T_{3}\right| \geq 2\left|T_{2}\right| \geq 2^{2} \\
\left|T_{4}\right| \geq 2\left|T_{3}\right| \geq 2^{3} \\
\ldots \\
\left|T_{n}\right| \geq 2\left|T_{n-1}\right| \geq 2^{n-1}
\end{gathered}
$$

This proves the theorem.

### 2.4 Bounding the topswops problem

Suppose, there is a topswop tree $T_{i}$ containing a leaf which needs the maximal number of iterations to terminate by topswop function. We can define $T_{i}$ as the tallest tree in the forest. We can regard the problem of finding the maximal height of a tree as finding the maximal number of flips for the topswop problem.
In this section we will develop a linear lower bound for the topswops problem by studying the specific number of flips of a particular family of sequences. A quadratic lower bound proven by Morales and Sudborough [8] and a Fibonacci upper bound proven by Klamkin [6] will also be shown in the section.

### 2.4.1 The linear lower bound

We denote $\Pi=\left\{\pi_{n}\right\}$ as an infinite family of permutations, where $\pi_{n}$ is a permutation on the integers $(1,2, \ldots, n)$. We will study on a particular family of permutations $\Gamma=\left\{\gamma_{n}\right\}$ in $\Pi$, where $n \geq 6$ and the permutations $\gamma_{n}$ are defined by:

$$
\gamma_{n}=3,1,4,5,6, \ldots, n-1, n, 2
$$

We use run $\left(\gamma_{n}\right)$ to denote the sequence of iterates of topswop function and $\mid$ run $\left(\gamma_{n}\right) \mid$ denotes the length of this sequence.

Example 6. Here, we give an example when $n$ is even. When $n=8, \gamma_{8}=(3,1,4,5,6,7,8,2)$ and the iterates of $\gamma_{8}$ :

$$
\begin{aligned}
\gamma_{8}= & (3,1,4,5,6,7,8,2) \\
P_{4}= & (4,1,3,5,6,7,8,2) \\
& (5,3,1,4,6,7,8,2) \\
P_{6}= & (6,4,1,3,5,7,8,2) \\
& (7,5,3,1,4,6,8,2) \\
P_{8}=Q_{8}= & (8,6,4,1,3,5,7,2) \\
& (2,7,5,3,1,4,6,8) \\
& (7,2,5,3,1,4,6,8) \\
Q_{6}= & (6,4,1,3,5,2,7,8) \\
& (2,5,3,1,4,6,7,8) \\
& (5,2,3,1,5,6,7,8) \\
Q_{4}= & (4,1,3,2,5,6,7,8) \\
& (2,3,1,4,5,6,7,8) \\
& (3,2,1,4,5,6,7,8) \\
& (1,2,3,4,5,6,7,8)
\end{aligned}
$$

where $P$ are the iterations of permutations from $(4,1,3,5,6,7,8,2)$ to $(8,6,4,1,3,5,7,2)$ and $Q$ are the iterations of permutations from $(8,6,4,1,3,5,7,2)$ to $(4,1,3,2,5,6,7,8)$.

We show the iterates of $\gamma_{8}$ by topswop function:

$$
\begin{array}{r}
f\left(\gamma_{8}\right)=P_{4} \\
f^{(2)}\left(P_{4}\right)=P_{6} \\
f^{(2)}\left(P_{6}\right)=P_{8}=Q_{8} \\
f^{(3)}\left(Q_{8}\right)=Q_{6} \\
f^{(3)}\left(Q_{6}\right)=Q_{4} \\
f^{(3)}\left(Q_{4}\right)=(1,2,3,4,5,6,7,8)
\end{array}
$$

We can derive $\left|\operatorname{run}\left(\gamma_{8}\right)\right|=1+2 * 2+3 * 2+3=14$.
We give another example when $n$ is odd. When $n=7, \gamma_{7}=(3,1,4,5,6,7,2)$ and the iterates of $\gamma_{7}$ :

$$
\begin{aligned}
\gamma_{7}=P_{3}= & (3,1,4,5,6,7,2) \\
& (4,1,3,5,6,7,2) \\
P_{5}= & (5,3,1,4,6,7,2) \\
& (6,4,1,3,5,7,2) \\
P_{7}=Q_{7}= & (7,5,3,1,4,6,2) \\
& (2,6,4,1,3,5,7) \\
& (6,2,4,1,3,5,7) \\
Q_{5}= & (5,3,1,4,2,6,7) \\
& (2,4,1,3,5,6,7) \\
& (4,2,1,3,5,6,7) \\
Q_{3}= & (3,1,2,4,5,6,7) \\
& (2,1,3,4,5,6,7) \\
& (1,2,3,4,5,6,7)
\end{aligned}
$$

where $P$ are the iterations of permutations from $\gamma_{7}$ to $(7,5,3,1,4,6,2)$ and $Q$ are the iterations of permutations from $(7,5,3,1,4,6,2)$ to $(3,1,2,4,5,6,7)$. We show the iterates of $\gamma_{8}$ by topswop function:

$$
\begin{array}{r}
f^{(2)}\left(\gamma_{7}\right)=P_{5} \\
f^{(2)}\left(P_{5}\right)=P_{7}=Q_{7} \\
f^{(3)}\left(Q_{7}\right)=Q_{5} \\
f^{(3)}\left(Q_{5}\right)=Q_{3} \\
f^{(2)}\left(Q_{3}\right)=(1,2,3,4,5,6,7)
\end{array}
$$

We can derive $\left|\operatorname{run}\left(\gamma_{7}\right)\right|=2 * 2+3 * 2+2=12$.

Theorem 3. For, $n \geq 6,\left|\operatorname{run}\left(\gamma_{n}\right)\right|$ is equal to $\frac{5 n}{2}-6$ when $n$ is even and $\frac{5 n}{2}-\frac{11}{2}$ when $n$ is odd.

Proof. The case when $n$ is even:
We define $i$ as an even number and let $6 \leq i \leq n$ and let $P$ represents the iterations of permutations from $P_{4}$ to $P_{n}$, where $P_{4}=(4,1,3,5,6, \ldots, n, 2)$ and $P_{n}=(n, n-$ $2, \ldots, 4,1,3, \ldots, n-3, n-1, n, 2)$. Consider the permutation with $n$ elements:

$$
P_{i-2}=(i-2, i-4, \ldots, 4,1,3, \ldots i-3, i-1, i, i+1, i+2, \ldots, n, 2)
$$

we iterate it twice:

$$
\begin{aligned}
& (i-1, i-3, \ldots, 3,1,4, \ldots i-4, i-2, i, i+1, i+2, \ldots, n, 2) \\
& P_{i}=(i, i-2, \ldots, 4,1,3, \ldots i-3, i-1, i+1, i+2, \ldots, n, 2)
\end{aligned}
$$

By the topswop function, we have $f^{(2)}\left(P_{i-2}\right)=P_{i}$.
Then, we define $j$ as an even number and let $6 \leq j \leq n$ and let $Q$ represents the iterations of permutations from $Q_{n}$ to $Q_{4}$, where $Q_{n}=P_{n}$ and $Q_{4}=(4,1,3,2,5,6, \ldots, n-1, n)$. Consider the permutation with $n$ elements:

$$
Q_{j}=(j, j-2, j-4, \ldots, 4,1,3, \ldots j-3, j-1,2, j+1, j+2, \ldots, n)
$$

we iterate it three times:

$$
\begin{array}{r}
(2, j-1, j-3, \ldots, 3,1,4, \ldots j-4, j-2, j, j+1, j+2, \ldots, n) \\
(j-1,2, j-3, \ldots, 3,1,4, \ldots j-4, j-2, j, j+1, j+2, \ldots, n) \\
Q_{j-2}=(j-2, j-4, \ldots, 4,1,3, \ldots j-3,2, j-1, j+1, j+2, \ldots, n)
\end{array}
$$

By the topswop function, we have $f^{(3)}\left(Q_{j}\right)=Q_{j-2}$.
We show the iterations of $\gamma_{n}$ by topswop functions:

$$
\begin{array}{r}
f\left(\gamma_{n}\right)=P_{4} \\
f^{(2)}\left(P_{4}\right)=P_{6} \\
f^{(2)}\left(P_{6}\right)=P_{8} \\
\ldots \\
f^{(2)}\left(P_{n-2}\right)=P_{n}=Q_{n} \\
f^{(3)}\left(Q_{n}\right)=Q_{n-2} \\
f^{(3)}\left(Q_{n-2}\right)=Q_{n-4} \\
\ldots \\
f^{(3)}\left(Q_{6}\right)=Q_{4} \\
f^{(3)}\left(Q_{4}\right)=(1,2,3,4, \ldots, n)
\end{array}
$$

It takes $\left(\frac{n-4}{2}\right) 2+1=n-3$ iterations from $\gamma_{n}$ to $P_{n}$ and $\left(\frac{n-4}{2}\right) 3+3=\frac{3 n}{2}-3$ iterations from $Q_{n}$ to $(1,2,3, \ldots, n)$. There are total $(n-3)+\left(\frac{3 n}{2}-3\right)=\frac{5 n}{2}-6$ iterations from $\gamma_{n}$
to $(1,2,3, \ldots, n)$.

Now, we study on the case when $n$ is odd:
We define $i$ as an odd number and let $7 \leq i \leq n$ and let $P$ represents the iterations of permutations from $\gamma_{n}$ to $P_{n}$, where $P_{n}=(n, n-2, \ldots, 4,1,3, \ldots, n-3, n-1, n, 2)$.

Consider the permutation with $n$ elements:

$$
P_{i-2}=(i-2, i-4, \ldots, 3,1,4, \ldots i-3, i-1, i, i+1, i+2, \ldots, n, 2)
$$

we iterate it twice:

$$
\begin{aligned}
& (i-1, i-3, \ldots, 4,1,3, \ldots i-4, i-2, i, i+1, i+2, \ldots, n, 2) \\
P_{i}= & (i, i-2, i-4, \ldots, 3,1,4, \ldots i-3, i-1, i+1, i+2, \ldots, n, 2)
\end{aligned}
$$

By the topswop function, we have $f^{(2)}\left(P_{i-2}\right)=P_{i}$.
Then, we define $j$ as an even number and let $3 \leq j \leq n$ and let $Q$ represents the iterations of permutations from $Q_{n}$ to $Q_{3}$, where $Q_{n}=P_{n}$ and $Q_{3}=(3,1,2,4,5, \ldots, n-1, n)$. Consider the permutation with $n$ elements:

$$
Q_{j}=(j, j-2, j-4, \ldots, 3,1,4, \ldots j-3, j-1,2, j+1, j+2, \ldots, n)
$$

we iterate it three times:

$$
\begin{array}{r}
(2, j-1, j-3, \ldots, 4,1,3, \ldots j-4, j-2, j, j+1, j+2, \ldots, n) \\
(j-1,2, j-3, \ldots, 4,1,3, \ldots j-4, j-2, j, j+1, j+2, \ldots, n) \\
Q_{j-2}=(j-2, j-4, \ldots, 3,1,4, \ldots j-3,2, j-1, j+1, j+2, \ldots, n)
\end{array}
$$

By the topswop function, we have $f^{(3)}\left(Q_{j}\right)=Q_{j-2}$.
We show the iterations of $\gamma_{n}$ by topswop functions:

$$
\begin{array}{r}
f^{(2)}\left(\gamma_{n}\right)=P_{5} \\
f^{(2)}\left(P_{5}\right)=P_{7} \\
f^{(2)}\left(P_{7}\right)=P_{9} \\
\ldots \\
f^{(2)}\left(P_{n-2}\right)=P_{n}=Q_{n} \\
f^{(3)}\left(Q_{n}\right)=Q_{n-2} \\
f^{(3)}\left(Q_{n-2}\right)=Q_{n-4} \\
f^{(3)}\left(Q_{5}\right)=Q_{3} \\
f^{(2)}\left(Q_{3}\right)=(1,2,3,4, \ldots, n)
\end{array}
$$

It takes $\left(\frac{n-3}{2}\right) 2=n-3$ iterations from $\gamma_{n}$ to $P_{n}$ and $\left(\frac{n-3}{2}\right) 3+2=\frac{3 n}{2}-\frac{5}{2}$ iterations from $Q_{n}$ to $(1,2,3, \ldots, n)$. There are total $(n-3)+\left(\frac{3 n}{2}-\frac{5}{2}\right)=\frac{5 n}{2}-\frac{11}{2}$ iterations from
$\gamma_{n}$ to $(1,2,3, \ldots, n)$.

Since $\frac{5 n}{2}-\frac{11}{2}>\frac{5 n}{2}-6$, we can derive a linear lower bound for the topswop function which is $\frac{5 n}{2}-\frac{11}{2}$, when $n \geq 6$.

### 2.4.2 The best proven quadratic lower bound

We will start out by looking at an infinite family of permutations $\Pi=\left\{\pi_{n}\right\}$. The goal is to find a positive $d$ and let $\left|\operatorname{run}\left(\pi_{n}\right)\right| \geq d * n^{2}$. For each integer $1<k<n$, we denote $\Pi^{(k)}$ as the infinite family of permutations containing all permutations $\pi$ on $(1,2, \ldots, n)$ such that $\pi(j)=j$, for all $2 \leq j \leq n-k, k=8,16, \ldots$.

We are particularly interested in finding permutations in $\Pi^{(k)}$ whose fixed point is the identity permutation. Such a family is $\Sigma=\left\{\sigma_{n}\right\}$ in $\Pi^{(8)}$, where $n>17$ and the permutations $\sigma_{n}$ are defined by:
$\sigma_{n}=n,(2,3, \ldots, n-8), n-5, n-6, n-2, n-7,1, n-3, n-1, n-4$.
An example of a permutation in $\Sigma$ is $\sigma_{26}=26,(2,3, \ldots, 18), 21,20,24,19,1,23,25,22$. We can now find the lower bound by studying the number of iterations of $\sigma_{n}$. The following results are proved by Morales and Sudborough [8].

Lemma 5. For all $n \geq 24,\left|\operatorname{run}\left(\sigma_{n}\right)\right| \geq n / 5$.
Lemma 5 shows that the number of iterations of the family $\left\{\sigma_{n}\right\}$ has a linear lower bound.

Lemma 6. For all $n \geq 18$, such that $n \equiv 2(\bmod 8)$, $\operatorname{run}\left(\sigma_{n}\right)$ ends with the identity permutation.

In order to derive the quadratic lower bound, Morales and Sudborough define a chaining technique in the family $\left\{\sigma_{n} \mid n \geq 18\right.$ and $\left.n \equiv 2(\bmod 8)\right\}$ to create a family of permutations $\Pi$, where $\left|\operatorname{run}\left(\pi_{n}\right)\right| \geq d * n^{2}$ and $d>0$. For a permutation $\pi_{n} \in \Pi^{(t)}$ and $\pi_{n+k} \in \Pi^{(k)}$, we define the permutation $\pi_{n} \oplus \pi_{n+k}$ in $\Pi^{(t+k)}$ by

$$
\pi_{n} \oplus \pi_{n+k}[i]= \begin{cases}\pi_{n}[i], & \text { if } 1 \leq i \leq n \text { and } \pi_{n}[i] \neq 1,  \tag{2.3}\\ \pi_{n+k}[1], & \text { if } \pi_{n}[i]=1 \\ \pi_{n+k}[i], & \text { if } n+1 \leq i \leq n+k\end{cases}
$$

For example, for $\sigma_{26}=26,(2,3, \ldots, 18), 21,20,24,19,1,23,25,22$ and $\sigma_{34}=34,(2,3, \ldots, 26)$, $29,28,32,27,1,31,33,30$. We will derive: $\sigma_{26} \oplus \sigma_{34}=26,(2,3, \ldots, 18), 21,20,24,19,34,23$, $25,22,29,28,32,27,1,31,33,30$.

Lemma 7. For any $t>0$ and any permutations $\pi_{n}$ in $\Pi^{(t)}$ and $\pi_{n+k}$ in $\Pi^{(k)}$, such that $\pi_{n}$ terminates with the identity permutation, $\pi_{n} \oplus \pi_{n+k}$ is a permutation on $n+k$ symbols in $\Pi^{(t+k)}$ such that $\left|\operatorname{run}\left(\pi_{n} \oplus \pi_{n+k}\right)\right|=\left|\operatorname{run}\left(\pi_{n}\right)\right|+\left|\operatorname{run}\left(\pi_{n+k}\right)\right|$.

The chaining of permutations can also be applied to more than two permutations. For permutations $\sigma_{n}, \sigma_{n+k}, \ldots, \sigma_{n+m k}$, for $m \geq 1, \sigma_{n} \oplus \sigma_{n+k} \oplus \ldots \oplus \sigma_{n+m k}$ denote the permutation $\left(\ldots\left(\left(\sigma_{n} \oplus \sigma_{n+k}\right) \oplus \sigma_{n+2 k}\right) \ldots \oplus \sigma_{n+m k}\right)$.

Let $\pi_{26+8 m}=\sigma_{26} \oplus \sigma_{34} \oplus \sigma_{42} \ldots \oplus \sigma_{26+8 m}$, where $m \geq 1$. By Lemma 7 we can derive $\left|\operatorname{run}\left(\pi_{26+8 m}\right)\right|=\sum_{i=0}^{m}\left|\operatorname{run}\left(\sigma_{26+8 i}\right)\right|$. By Lemma 5 we can know $\mid$ run $\left(\sigma_{26+8 i}\right) \mid \geq$ $(26+8 i) / 5$, where $i \geq 0$. Thus, we can derive $\left|\operatorname{run}\left(\sigma_{26+8 m}\right)\right| \geq \sum_{i=0}^{m}(26+8 i) / 5 \geq$ $4 / 5 m^{2}+6 m+26 / 5$ and we can write it as the following corollary.

Corollary 7.1. For all $m \geq 1, \pi_{26+8 m}=\sigma_{26} \oplus \sigma_{34} \oplus \sigma_{42} \ldots \oplus \sigma_{26+8 m}$ is a permutation on $26+8 m$ symbols with $\left|\operatorname{run}\left(\pi_{26+8 m}\right)\right| \geq 4 / 5 m^{2}+6 m+26 / 5$.

Hence, a quadratic lower bound for the $\Pi$-family has been derived. By the above corollary, we can establish the following theorem.

Theorem 4. The topswop problem has a quadratic lower bound.

### 2.4.3 The Wilf upper bound

We will briefly introduce the Wilf upper bound proved by Wilf [4]. Let $P \in S_{n}$ and denote $P[i]$ as the $i$ th number of the permutation, where $1 \leq i \leq n$. A number is at the original position if $P[i]=i$.

Theorem 5. After each iteration of the topswop function, the Wilf number increases.
Proof. We perform one iteration of the topswop function. Each number at the original position and larger than $P[1]$, leaves the Wilf number unchanged. The remaining numbers at the original position will in general not be at the original position anymore. Nevertheless, the $P[1]$ 's number is at the correct position. And since the sum of the first $P[1]-1$ Wilf number is always smaller than the Wilf number of $P[1]$, the total Wilf number always increases with at least 1 for each iteration. (A power of two is larger than the sum of all earlier powers of two by exactly one unit, a fact which is the basis of binary counting).

The maximal Wilf number is derived when every number is at the original position. So the maximal Wilf number is $2^{n+1}-1$ and $|\operatorname{run}(P)| \leq 2^{n+1}-1$. Thus an exponential upper bound has been derived.

### 2.4.4 The Fibonacci upper bound

We now show a Fibonacci upper bound proven by Klamkin [6]. Suppose that during the algorithm, there are in total $k$ distinct values for $P[1]$, where $1 \leq k \leq n$. A Fibonacci number is denoted by $F_{i}$, where $F_{i}=F_{i-1}+F_{i-2}, F_{0}=0$ and $F_{1}=1$.

Theorem 6. When $P[1]$ takes on $k$ distinct values, $|\operatorname{run}(P)| \leq F_{k+1}$.
Proof. We give a proof by induction on $k$.

Base case: Show that the statement holds for the smallest value of $k=1$. For $k=1, P[1]=1$ and the algorithm directly terminates. We have $|\operatorname{run}(P)|=0$ and $F_{k+1}=F_{2}=1$. Thus $|\operatorname{run}(P)| \leq F_{k+1}$.

Inductive Step: Show that for any $k \geq 1$, if $|\operatorname{run}(P)| \leq F_{k+1}$ holds when $P$ [1] takes on $k$ distinct values, then $|\operatorname{run}(P)| \leq F_{k+2}$ also holds when $P[1]$ takes on $k+1$ distinct values. All $k+1$ values that $P[1]$ takes on, are ordered and can be written as: $d_{1} \leq \ldots \leq d_{k} \leq d_{k+1}$, where $d_{k+1}$ is the largest value. Suppose at the $r t h$ iteration we have $P[1]=d_{k+1}$. Denote $t=P\left[d_{k+1}\right]$, at the $(r+1)$ th iteration, we have $P[1]=t$ and $P\left[d_{k+1}\right]=d_{k+1}$. The remaining iterations will always retain $P\left[d_{k+1}\right]=d_{k+1}$.

Now, suppose if $t=1$. Then $P[1]=1$ at the $(r+1) t h$ iteration and the algorithm terminates. During the algorithm, we are sure that both $d_{k+1}$ and $t$ have never been at position $P[1]$. Thus $P[1]$ can take on at most $k$ distinct values $\left(d_{2}, d_{3}, \ldots, d_{k+1}\right)$. Then $r \leq F_{k+1}$ and we can get $|\operatorname{run}(P)|=r+1 \leq F_{k+1}+1 \leq F_{k+2}$.
Suppose if $t>1, P[1]$ can take on at most $k-1$ distinct values $\left(d_{2}, d_{3}, \ldots, d_{k+1}\right)$. Then $r \leq F_{k}$. By the induction assumption, we can get $|\operatorname{run}(P)| \leq F_{k+1}+r \leq F_{k+1}+F_{k}=$ $F_{k+2}$

Since both the base case and the inductive step have been proved as true, by mathematical induction the statement $|\operatorname{run}(P)| \leq F_{k+1}$ holds.

Suppose $P[1]$ takes on all $N$ values we can get $|\operatorname{run}(P)| \leq F_{N+1}$. Thus, we get the Fibonacci upper bound. By the asymptotic behaviour of Fibonacci sequence, we can know Fibonacci upper bound is an exponential upper bound.

## 3 Discussion

Morales and Sudborough show a quadratic lower bound while Klamkin shows a Fibonacci upper bound for the maximal height of the trees in the forest. As we can see from the following graph. There is a huge discrepancy between lower and upper bound. The quadratic lower bound is closer to the real values than the Fibonacci upper bound.


Figure 4: The relation between the maximum number of iterations and the length of the row in a semi-logarithmic graph. From "Topswops", Wikipedia, The Free Encyclopedia, 14 January 2021.

Morales and Sudborough derived the lower bound by studying on a specific tree while Klamkin derived the upper bound by using mathematical induction on the topswop algorithm. It seems we could get a bound closer to the real values by finding a proper tree. One possible way is to find a family of permutation which ends with the identity permutation and apply the chaining techniques introduced by Morales and Sudborough. And we may derive a better upper bound by finding the tallest tree.

It is also difficult to produce the exact number of nodes for each topswop tree when $n \geq 12$. The biggest reason is that the number of nodes in the forest has a factorial growth. When $n=12$ there are around 5 billion nodes in the forest. Thus, it could be quite time-consuming for finding the size of each tree.

## References

[1] D. Berman and M. S. Klamkin. A reverse card shuffle. SIAM Review, 18(3):491-492, 1976.
[2] B Bollobas. Modern Graph Theory. Graduate Texts in Mathematics 184. SpringerVerlag New York, 1 edition, 1998.
[3] J. Cibulka. On average and highest number of flips in pancake sorting. Theoret. Comput. Sci., 412(8-10):822-834, 2011.
[4] M. Gardner. Time Travel and Other Mathematical Bewilderments. W.H. Freeman, 1988.
[5] K. Kimura, A. Takahashi, T. Araki, and K. Amano. Maximum number of steps of topswops on 18 and 19 cards. CoRR, abs/2103.08346, 2021.
[6] M. S. Klamkin. Problems in Applied Mathematics: Selections from SIAM Review. Society for Industrial and Applied Mathematics, 1990.
[7] D. J. Kleitman, E. Kramer, J. H. Conway, S. Bell, and H. Dweighter. Problems and Solutions: Elementary Problems: E2564-E2569. Amer. Math. Monthly, 82(10):1009-1010, 1975.
[8] L. Morales and H. Sudborough. A quadratic lower bound for topswops. Theoret. Comput. Sci., 411(44-46):3965-3970, 2010.
[9] S. Rabinowitz and M. Bowron. Index to Mathematical Problems, 1975-1979. Indexes to mathematical problems. MathPro Press, 1999.
[10] N. Sloane. The on-line encyclopedia of integer sequences. volume 1, page 130, 01 2007.

## Appendices

## A Tables for the number of nodes and the largest number of flips

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| 1,2 | 2 | 1 |

Table 2: $n=2$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3$ | 5 | 2 |
| $1,3,2$ | 1 | 0 |

Table 3: $n=3$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4$ | 12 | 4 |
| $1,3,2,4$ | 6 | 3 |
| $1,4,3,2$ | 2 | 1 |
| $1,2,4,3$ | 2 | 1 |
| $1,4,2,3$ | 1 | 0 |
| $1,3,4,2$ | 1 | 0 |

Table 4: $n=4$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5$ | 34 | 7 |
| $1,4,3,2,5$ | 18 | 5 |
| $1,3,2,4,5$ | 18 | 6 |
| $1,4,2,3,5$ | 7 | 4 |
| $1,2,3,5,4$ | 5 | 2 |
| $1,5,3,4,2$ | 4 | 2 |

Table 5: $n=5$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6$ | 108 | 10 |
| $1,3,2,4,5,6$ | 69 | 9 |
| $1,4,3,2,5,6$ | 57 | 10 |
| $1,5,4,3,2,6$ | 35 | 8 |
| $1,5,4,2,3,6$ | 19 | 7 |
| $1,5,3,4,2,6$ | 19 | 5 |

Table 6: $n=6$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6,7$ | 407 | 16 |
| $1,3,2,4,5,6,7$ | 271 | 14 |
| $1,4,3,2,5,6,7$ | 198 | 15 |
| $1,5,4,3,2,6,7$ | 116 | 10 |
| $1,6,5,4,3,2,7$ | 115 | 11 |
| $1,4,2,3,5,6,7$ | 101 | 13 |

Table 7: $n=7$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6,7,8$ | 1867 | 22 |
| $1,3,2,4,5,6,7,8$ | 1097 | 20 |
| $1,4,3,2,5,6,7,8$ | 999 | 17 |
| $1,6,5,4,3,2,7,8$ | 506 | 18 |
| $1,5,2,3,4,6,7,8$ | 490 | 18 |
| $1,4,2,3,5,6,7,8$ | 442 | 17 |

Table 8: $n=8$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6,7,8,9$ | 9718 | 30 |
| $1,3,2,4,5,6,7,8,9$ | 5583 | 25 |
| $1,4,3,2,5,6,7,8,9$ | 4587 | 27 |
| $1,5,4,3,2,6,7,8,9$ | 2107 | 19 |
| $1,5,2,3,4,6,7,8,9$ | 2092 | 24 |
| $1,6,5,4,3,2,7,8,9$ | 1953 | 21 |

Table 9: $n=9$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6,7,8,9,10$ | 62200 | 38 |
| $1,3,2,4,5,6,7,8,9,10$ | 33093 | 34 |
| $1,4,3,2,5,6,7,8,9,10$ | 25621 | 35 |
| $1,3,4,2,5,6,7,8,9,10$ | 11883 | 30 |
| $1,5,4,3,2,6,7,8,9,10$ | 11096 | 28 |
| $1,5,4,2,3,6,7,8,9,10$ | 10911 | 30 |

Table 10: $n=10$

| Root | Number of nodes | Largest number of flips |
| :---: | :---: | :---: |
| $1,2,3,4,5,6,7,8,9,10,11$ | 440330 | 51 |
| $1,3,2,4,5,6,7,8,9,10,11$ | 232038 | 44 |
| $1,4,3,2,5,6,7,8,9,10,11$ | 160618 | 43 |
| $1,3,4,2,5,6,7,8,9,10,11$ | 77339 | 36 |
| $1,6,5,4,3,2,7,8,9,10,11$ | 68686 | 40 |
| $1,5,4,3,2,6,7,8,9,10,11$ | 68370 | 36 |

Table 11: $n=11$

## B Regression results

$\left\{\begin{array}{ll|llll} & & \text { Estimate } & \text { SE } & \text { TStat } & \text { PValue } \\\right.$\cline { 2 - 6 } \& ParameterTable$\rightarrow & -0.27867 & 0.0919026 & -3.03223 & 0.0190546 \\ & \mathrm{x} & 0.378856 & 0.0316468 & 11.9714 & 6.46092 \times 10^{-6}, \\ & \mathrm{x}^{2} & 0.0750753 & 0.00238954 & 31.4183 & 8.54962 \times 10^{-9}\end{array}$

| RSquared $\rightarrow 0.999863$, | AdjustedRSquared $\rightarrow 0.999824, ~ E s t i m a t e d V a r i a n c e ~$ |
| ---: | :--- |$\rightarrow 0.00301483$,

Table 12: Regression result for the yellow line: family of the permutation $(1,2,3,4, \ldots, n)$. The regression model: $a+b n+c n^{2}$.

We first do the F-test: we test the null hypothesis,

$$
H_{0}: a=b=c=0
$$

versus the alternative

$$
H_{a}: \text { at least one of the coefficients } a, b, c \text { is non-zero }
$$

We choose $\alpha=0.01$ as the significant level. The $P$ value is 0.000 which is smaller than 0.01 , we can reject the null hypothesis. There is a significant relationship between the number of nodes and the length of the permutation.

Then we do the t-test: The null and alternative hypotheses for a hypotheses test about the coefficient $b$ are written as

$$
\begin{aligned}
& H_{0}: b=0 \\
& H_{a}: b \neq 0
\end{aligned}
$$

We choose $\alpha=0.01$ as the significant level. The $P$ value is 0.000 which is smaller than 0.01 , we can reject the null hypothesis and the coefficient $b$ is significant.

Now, we do the t-test for the coefficient $c$. The null and alternative hypotheses for a hypotheses test about the coefficient $b$ are written as

$$
\begin{aligned}
& H_{0}: c=0 \\
& H_{a}: c \neq 0
\end{aligned}
$$

We choose $\alpha=0.01$ as the significant level. The $P$ value is 0.000 which is smaller than 0.01 , we can reject the null hypothesis and the coefficient $c$ is significant. Since $c$ is significant, we can say there is more than an exponential growth for the number of nodes.

## C Codes: Compute number of nodes for each tree

```
f[arrO_, n0_] := Module[{i, temp, arr = arr0, n = n0, index = n0},
    For[i = 1, i <= Floor[n/2], i++,
    temp = arr[[i]];
    arr[[i]] = arr[[index]];
    arr[[index]] = temp;
    index -= 1;
    ];
    Return[arr]
    ]
(*number of nodes for each tree*)
NumOfNodes[list0_] := Module[{len, leafs, i, same, j, list},
    list = List[list0];
        len = Length[list[[1]]];
    For[i = 1, i <= Length[list], i++,
        For[j = 2, j <= len, j++,
            If[list[[i, j]] == j, AppendTo[list, f[list[[i]] , j ]] ];
            ];
        ];
    Return[Length[list]]
    ]
```


## D Codes: Compute number of flips for the specific tree

(*The number of flips for root $(1,2, \ldots . \ldots) *)$

```
list = List[{1, 2, 3, 4, 5}];
len = Length[list[[1]]];
For[i = 1, i <= Length[list], i++,
    For[j = 2, j <= len, j++,
        If[list[[i, j]] == j, AppendTo[list, f[list[[i]] , j ]] ];
        ];
    ]
arr = list[[-1]];
index = 0;
While[arr[[1]] != 1, index++; temp = prefixRev[arr]; arr = temp;
    Print[temp] ]
```

index

## E Codes: Compute the maximum number of flips in the topswop forest

```
f[arr0_, n0_] := Module[{i, temp, arr = arr0, n = n0, index = n0},
    For[i = 1, i <= Floor[n/2], i++,
    temp = arr[[i]];
    arr[[i]] = arr[[index]];
    arr[[index]] = temp;
    index -= 1;
    ];
    Return[arr]
    ]
```

(*
3.1 input a vetctor(v1,v2..vn) and output prefix reversal
*)
prefixRev[arr0_] := Module[\{n0, arr\},
$\mathrm{n} 0=\operatorname{arr} 0[[1]] ;$
arr $=\mathrm{f}[\operatorname{arr} 0, \mathrm{n} 0]$;
Return[arr];
]
(*
3.2 output the pancake graph Gn in matrix form.
*)
deteGraph[n0_] := Module[\{p, ma, fac, i, n, j\},
p = Permutations[Table[i, \{i, n0\}]];
fac = Factorial[n0];
ma $=$ Table[0, $\{x, f a c\},\{y, f a c\}] ;$
For $[i=1$, $i<=f a c, i++$,
For $[\mathrm{n}=1, \mathrm{n}<=\mathrm{n} 0, \mathrm{n}++$,
For $[j=1, j<=f a c, j++$,
$\operatorname{If}[\operatorname{prefixRev}[p[[i]]]==p[[j]], \operatorname{ma}[[i, j]]=1 ; \operatorname{Break}[] \quad$ ]
]
]
];
For $[i=1, i<=f a c, i++$,

```
        ma[[i, i]] = 0;
        ];
    Return[ma]
    ]
(*
3.3 input AdjacencyGraph and n, output the maximum number of \
flips
*)
findMaxNum[adjGraph_, n0_] :=
    Module[{permu, i, index, j, begin, end, leaf, step, max},
    permu = Permutations[Table[i, {i, n0}]];
    begin = Factorial[n0 - 1];
    end = Factorial[n0];
    leaf = List[];
    For[i = begin + 1, i <= end, i++,
        index = 0;
        For[j = 1, j <= n0, j++,
            If[permu[[ i, j ]] == j, Break[], index++ ]
        ]
        If[index == n0, leaf = Append[leaf, i]];
        ];
    max = 0;
    For[i = 1, i <= begin, i++,
        For[j = 1, j <= Length[leaf], j++,
        step = FindShortestPath[adjGraph, leaf[[j]], i];
        If[max < Length[step], max = Length[step]];
        ]
    ];
    Return[max - 1];
    ]
```

Lnu.se

## Linnæus University

Sweden
Faculty of Technology
SE-391 82 Kalmar | SE-351 95 Växjö
Phone +46 (0)772-28 8000
teknik@lnu.se
Lnu.se/faculty-of-technology?1=en

