Quantifying CDS Sortability of Permutations Using Strategic Piles

Context Directed Swag
M. Gaetz, B. Molokach, M. Scheepers, M. Shanks

July 29, 2016

Motivation

Ciliates

- Sorting is a fundamental to many natural, industrial, and commercial processes.

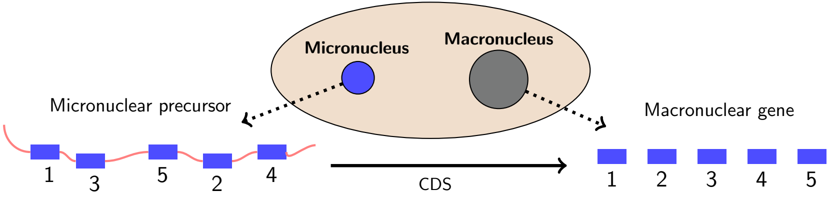


Figure: CDS as used in ciliates

Ciliates

- Sorting is a fundamental to many natural, industrial, and commercial processes.
- A natural example: the one-celled organisms known as a ciliates.

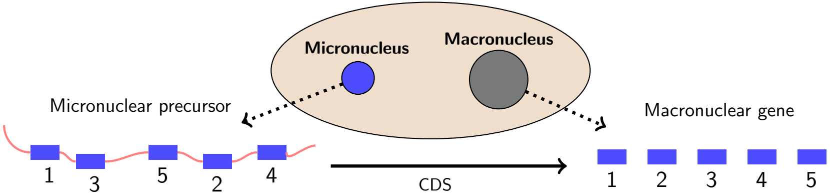


Figure: CDS as used in ciliates

Ciliates

- Sorting is a fundamental to many natural, industrial, and commercial processes.
- A natural example: the one-celled organisms known as a ciliates.
- A ciliate must sort DNA in its micronuclei in order to create a macronucleus.

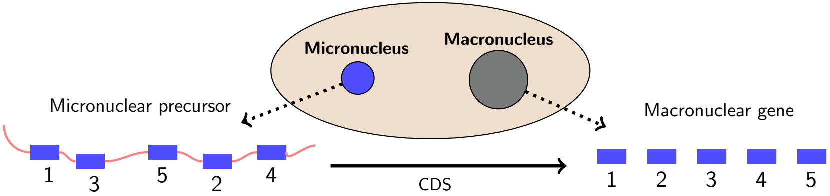


Figure: CDS as used in ciliates

Ciliates

- Sorting is a fundamental to many natural, industrial, and commercial processes.
- A natural example: the one-celled organisms known as a ciliates.
- A ciliate must sort DNA in its micronuclei in order to create a macronucleus.
- Biologists hypothesize that ciliates use iterations of the context directed swap (CDS) sorting operation to rearrange their DNA [3].

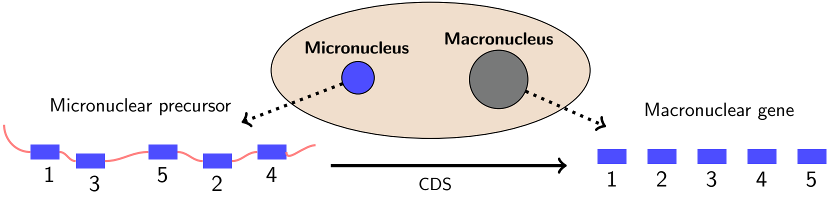


Figure: CDS as used in ciliates

Preliminaries

Context Directed Swaps

Consider $\pi \in S_n$ where $\pi = [a_1\ a_2\ \ldots\ a_n]$. To each a_i, assign a left pointer, $\langle a_i - 1, a_i \rangle$, and a right pointer, $\langle a_i, a_i + 1 \rangle$. The CDS operation uses two pointers p and q in the order $p \ldots q \ldots p \ldots q$ and exchanges the elements flanked by these pointers.

CDS Example

Consider the permutation $\pi = [2\ 4\ 3\ 5\ 1]$:

CDS Example

Consider the permutation $\pi = [2\ 4\ 3\ 5\ 1]$:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}, {}_{\color{blue}\langle 3,4\rangle}\ 4_{\color{red}\langle 4,5\rangle}, {}_{\langle 2,3\rangle}\ 3_{\color{blue}\langle 3,4\rangle}, {}_{\color{red}\langle 4,5\rangle}\ 5_{\langle 5,6\rangle}, {}_{\langle 0,1\rangle}\ 1_{\langle 1,2\rangle}]$$

CDS Example

Consider the permutation $\pi = [2\ 4\ 3\ 5\ 1]$:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}\,,_{\langle 3,4\rangle}\ 4_{\langle 4,5\rangle}\,,_{\langle 2,3\rangle}\ 3_{\langle 3,4\rangle}\,,_{\langle 4,5\rangle}\ 5_{\langle 5,6\rangle}\,,_{\langle 0,1\rangle}\ 1_{\langle 1,2\rangle}]$$

Using the pointers $p = \langle 3, 4\rangle$ and $q = \langle 4, 5\rangle$, we get:

CDS Example

Consider the permutation $\pi = [2\ 4\ 3\ 5\ 1]$:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}\ ,_{\langle 3,4\rangle} 4_{\langle 4,5\rangle}\ ,_{\langle 2,3\rangle} 3_{\langle 3,4\rangle}\ ,_{\langle 4,5\rangle} 5_{\langle 5,6\rangle}\ ,_{\langle 0,1\rangle} 1_{\langle 1,2\rangle}]$$

Using the pointers $p = \langle 3, 4\rangle$ and $q = \langle 4, 5\rangle$, we get:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}\ ,_{\langle 2,3\rangle} 3_{\langle 3,4\rangle}\ ,_{\langle 3,4\rangle} 4_{\langle 4,5\rangle}\ ,_{\langle 4,5\rangle} 5_{\langle 5,6\rangle}\ ,_{\langle 0,1\rangle} 1_{\langle 1,2\rangle}]$$

CDS Example

Consider the permutation $\pi = [2\ 4\ 3\ 5\ 1]$:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}\, ,_{\langle 3,4\rangle} 4_{\langle 4,5\rangle}\, ,_{\langle 2,3\rangle} 3_{\langle 3,4\rangle}\, ,_{\langle 4,5\rangle} 5_{\langle 5,6\rangle}\, ,_{\langle 0,1\rangle} 1_{\langle 1,2\rangle}]$$

Using the pointers $p = \langle 3, 4\rangle$ and $q = \langle 4, 5\rangle$, we get:

$$[_{\langle 1,2\rangle} 2_{\langle 2,3\rangle}\, ,_{\langle 2,3\rangle} 3_{\langle 3,4\rangle}\, ,_{\langle 3,4\rangle} 4_{\langle 4,5\rangle}\, ,_{\langle 4,5\rangle} 5_{\langle 5,6\rangle}\, ,_{\langle 0,1\rangle} 1_{\langle 1,2\rangle}]$$

There are no remaining pointers in the order $p \dots q \dots p \dots q$.

Strategic Piles

- CDS cannot sort every permutation.

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \dots q \dots p \dots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation,** C_π**,** of π:

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation,** C_π**,** of π:
  - Let $X = (0\ 1\ 2\ \ldots\ n)$

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation, C_π,** of π:
  - Let $X = (0\ 1\ 2\ \ldots\ n)$
  - Let $Y_\pi = (0\ a_n\ a_{n-1}\ \ldots\ a_2\ a_1)$

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation,** C_π**,** of π:
  - Let $X = (0\ 1\ 2\ \ldots\ n)$
  - Let $Y_\pi = (0\ a_n\ a_{n-1}\ \ldots\ a_2\ a_1)$
  - Then $C_\pi = Y_\pi \circ X$.

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation,** C_π**,** of π:
  - Let $X = (0\ 1\ 2\ \ldots\ n)$
  - Let $Y_\pi = (0\ a_n\ a_{n-1}\ \ldots\ a_2\ a_1)$
  - Then $C_\pi = Y_\pi \circ X$.
- If C_π contains a cycle of the form $= (0\ \ldots\ n\ b_1\ \ldots\ b_k)$, then $SP(\pi) = \{b_1, \ldots, b_k\}$. Otherwise, π is sortable and $SP(\pi) = \emptyset$.

Strategic Piles

- CDS cannot sort every permutation.
- When any rotation of the sorted permutation occurs, there are no remaining pointers in the order $p \ldots q \ldots p \ldots q$, and we call this a **fixed point**.
- We refer to each fixed point by the last number occurring in the permutation.
- The **strategic pile** of π, $SP(\pi)$, is the collection of fixed points of π, and can be found by using **the cycle permutation,** C_π**,** of π:
  - Let $X = (0\ 1\ 2\ \ldots\ n)$
  - Let $Y_\pi = (0\ a_n\ a_{n-1}\ \ldots\ a_2\ a_1)$
  - Then $C_\pi = Y_\pi \circ X$.
- If C_π contains a cycle of the form $= (0\ \ldots\ n\ b_1\ \ldots\ b_k)$, then $SP(\pi) = \{b_1, \ldots, b_k\}$. Otherwise, π is sortable and $SP(\pi) = \emptyset$.
- Often, our results depend on the order of the strategic pile elements, so assume $\{b_1, \ldots, b_k\}$ occur chronologically in C_π.

Strategic Pile Example

Consider the permutation $\pi = [4\ 2\ 1\ 3\ 5]$

Strategic Pile Example

Consider the permutation $\pi = [4\ 2\ 1\ 3\ 5]$

- $Y_\pi = (0\ 5\ 3\ 1\ 2\ 4)$

Strategic Pile Example

Consider the permutation $\pi = [4\ 2\ 1\ 3\ 5]$

- $Y_\pi = (0\ 5\ 3\ 1\ 2\ 4)$
- $X = (0\ 1\ 2\ 3\ 4\ 5)$

Strategic Pile Example

Consider the permutation $\pi = [4\ 2\ 1\ 3\ 5]$

- $Y_\pi = (0\ 5\ 3\ 1\ 2\ 4)$
- $X = (0\ 1\ 2\ 3\ 4\ 5)$
- $C_\pi = Y_\pi \circ X = (0\ 5\ 3\ 1\ 2\ 4)(0\ 1\ 2\ 3\ 4\ 5) = (0\ 2\ 1\ 4\ 3)(5)$

Strategic Pile Example

Consider the permutation $\pi = [4\ 2\ 1\ 3\ 5]$

- $Y_\pi = (0\ 5\ 3\ 1\ 2\ 4)$
- $X = (0\ 1\ 2\ 3\ 4\ 5)$
- $C_\pi = Y_\pi \circ X = (0\ 5\ 3\ 1\ 2\ 4)(0\ 1\ 2\ 3\ 4\ 5) = (0\ 2\ 1\ 4\ 3)(5)$

Since 0 and 5 are not in the same cycle, the permutation has empty strategic pile and so is sortable.

Research Goal

Our goal is to count the number of permutations in S_n that, when operated on by CDS, can result in k different fixed points. In a sense, we count the number of permutations with a specific size strategic pile in order to quantify permutations by how "close" they are to being sortable.

Full Strategic Pile

Full Strategic Piles

Definition

A permutation $\pi \in S_n$ has a *full strategic pile* if $|\mathrm{SP}(\pi)| = n - 1$.

Full Strategic Piles

Definition

A permutation $\pi \in S_n$ has a *full strategic pile* if $|\mathrm{SP}(\pi)| = n - 1$.

These permutations have the highest "degree of unsortability".

Full Strategic Piles

Definition

A permutation $\pi \in S_n$ has a *full strategic pile* if $|\mathrm{SP}(\pi)| = n - 1$.

These permutations have the highest "degree of unsortability".

We are interested in counting the number of permutations with full strategic pile.

Full Strategic Piles

Definition

A permutation $\pi \in S_n$ has a *full strategic pile* if $|\mathrm{SP}(\pi)| = n - 1$.

These permutations have the highest "degree of unsortability".

We are interested in counting the number of permutations with full strategic pile.

If $\pi \in S_n$ has a full strategic pile, then $C_\pi = (0\ n\ b_1\ \ldots\ b_{n-1})$.

Full Strategic Piles

Definition

A permutation $\pi \in S_n$ has a *full strategic pile* if $|\mathrm{SP}(\pi)| = n - 1$.

These permutations have the highest "degree of unsortability".

We are interested in counting the number of permutations with full strategic pile.

If $\pi \in S_n$ has a full strategic pile, then $C_\pi = (0\ n\ b_1\ \ldots\ b_{n-1})$.

Thus, we can count all C_π of this form.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Recall that $C_\pi = Y_\pi \circ X$.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Recall that $C_\pi = Y_\pi \circ X$.

Proof.

Assume C_π has a full strategic pile and n is odd. Then C_π is of length $n+1$, and the permutation is odd.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Recall that $C_\pi = Y_\pi \circ X$.

Proof.

Assume C_π has a full strategic pile and n is odd. Then C_π is of length $n+1$, and the permutation is odd. Since both X and Y_π are of length $n+1$, the are both odd permutations.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Recall that $C_\pi = Y_\pi \circ X$.

Proof.

Assume C_π has a full strategic pile and n is odd. Then C_π is of length $n+1$, and the permutation is odd. Since both X and Y_π are of length $n+1$, the are both odd permutations. Thus, two odd factors must yield a odd result, which is a contradiction.

Permutations with odd length

No permutation of odd length has a full strategic pile.

Theorem

If $\pi \in S_n$ has a full strategic pile, then n is even.

Recall that $C_\pi = Y_\pi \circ X$.

Proof.

Assume C_π has a full strategic pile and n is odd. Then C_π is of length $n+1$, and the permutation is odd. Since both X and Y_π are of length $n+1$, the are both odd permutations. Thus, two odd factors must yield a odd result, which is a contradiction. Therefore n is even. □

Main Result

Theorem

For an even n, the number of permutations in S_n with full strategic pile is $\frac{2(n-1)!}{n}$.

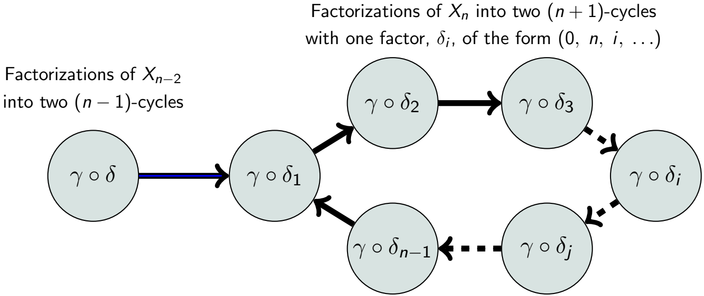


Figure: A graphical explanation of the different transformations that will yield the formula above.

A Known Counting Argument

We first must note that the number of factorizations of an $(n-1)$-cycle into two cycles of length $(n-1)$ is known.

A Known Counting Argument

We first must note that the number of factorizations of an $(n-1)$-cycle into two cycles of length $(n-1)$ is known.

Lemma (Cangelmi, [2])

Let $\sigma \in S_{n-1}$ be an even $(n-1)$-cycle. Then the number factorizations of σ into two $(n-1)$-cycles is $\frac{2(n-2)!}{n}$.

A Known Counting Argument

We first must note that the number of factorizations of an $(n-1)$-cycle into two cycles of length $(n-1)$ is known.

Lemma (Cangelmi, [2])

Let $\sigma \in S_{n-1}$ be an even $(n-1)$-cycle. Then the number factorizations of σ into two $(n-1)$-cycles is $\frac{2(n-2)!}{n}$.

Now we will start with two factors of X_{n-2} that are $(n-1)$-cycles and injectively map to all factorizations of X_n two $(n+1)$-cycles where the right-most factor is of the form $(0\ n\ \ldots)$.

First Transformation

Let $\lambda_n = (0\ n\ 1)$ and $c_n = (1\ 2\ \ldots\ n-1)$.

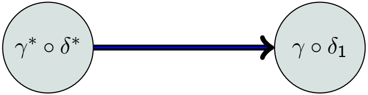
Lemma

Let $X_{n-2} = \gamma \circ \delta$ where γ and δ are $(n-1)$-cycles. We define the following:

$$\gamma_1 = \lambda_n \circ c_n \circ \gamma \circ (c_n)^{-1}$$
$$\delta_1 = c_n \circ \delta \circ (c_n)^{-1} \circ \lambda_n$$

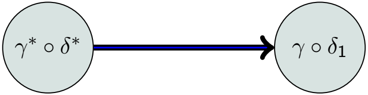
Then γ_1 and δ_1 are $(n+1)$-cycles, δ_1 is of the form $(0\ n\ 1\ \ldots)$, and $X_n = \gamma_1 \circ \delta_1$.

First Transformation Explanation



We have from [3] that there are $\frac{2(n-2)!}{n}$ factorizations on the left, and since the map defined is injective, there are this many on the right as well.

First Transformation Explanation



We have from [3] that there are $\frac{2(n-2)!}{n}$ factorizations on the left, and since the map defined is injective, there are this many on the right as well.

Note that we have only mapped to δ factors of the from $(0\ n\ 1\ \ldots)$, and now must retrieve other factors of the form $(0\ n\ i\ \ldots)$

Second Transformation

We now show that another injective map will allow for the inclusion of δ factors that are of the form $(0\ n\ i\ \ldots)$ for all $2 \leq i \leq n-1$.

Lemma

Assume γ_i and δ_i are $(n+1)$-cycles, δ_i is of the form $(0\ n\ i\ \ldots)$, and $\gamma_i \circ \delta_i = X_n$. Define:

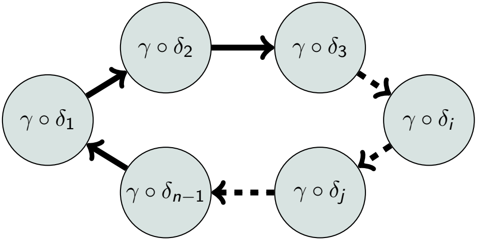
$$\gamma_{i+1} = r_n \circ c_n \circ \gamma_i \circ (c_n)^{-1}$$
$$\delta_{i+1} = c_n \circ \delta_i \circ (c_n)^{-1}$$

where $r_n = (2\ 1\ n)$. Then, γ_{i+1} and δ_{i+1} are $(n+1)$-cycles, δ_{i+1} is of the form $(0\ n\ i+1\ \ldots)$, and $\gamma_{i+1} \circ \delta_{i+1} = X_n$.

Lemma

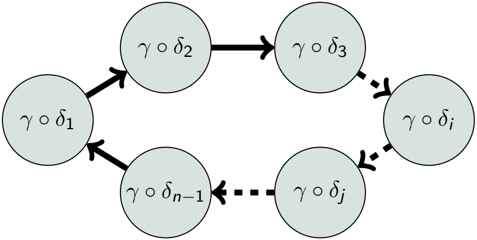
For any $1 < i < n$, $\delta_i \neq \delta_1$, and $\delta_n = \delta_1$.

Second Transformation Explanation



Since all maps are injective, there are still $\frac{2(n-2)!}{n}$ factorizations in each circle. Also, all of these factors are distinct.

Second Transformation Explanation



Since all maps are injective, there are still $\frac{2(n-2)!}{n}$ factorizations in each circle. Also, all of these factors are distinct.

The third Lemma stated shows that after this map is applied the $(n-1)^{st}$ time, the result will return to circle holding the δ_1 factorizations.

Theorem Proof

Proof sketch:

Theorem Proof

Proof sketch:

- To prove the theorem, we must show that these three Lemmas have counted exactly all of the δ_i factors with the desired form.

Theorem Proof

Proof sketch:

- To prove the theorem, we must show that these three Lemmas have counted exactly all of the δ_i factors with the desired form.
- To do this, we use the fact that permutations are invertible maps to go from these δ_i factors of X_n back to the δ factor of X_{n-2}.

Theorem Proof

Proof sketch:

- To prove the theorem, we must show that these three Lemmas have counted exactly all of the δ_i factors with the desired form.
- To do this, we use the fact that permutations are invertible maps to go from these δ_i factors of X_n back to the δ factor of X_{n-2}.
- This is sufficient to show that all the factorizations have been counted.

Theorem Proof

Proof sketch:

- To prove the theorem, we must show that these three Lemmas have counted exactly all of the δ_i factors with the desired form.
- To do this, we use the fact that permutations are invertible maps to go from these δ_i factors of X_n back to the δ factor of X_{n-2}.
- This is sufficient to show that all the factorizations have been counted.
- Thus there are a total of $n-1$ injective maps, all of whom have a size of $\frac{2(n-2)!}{n}$ [3]. So we multiply by $n-1$ to get the final count of $\frac{2(n-1)!}{n}$.

Strategic Piles of Size k

Form for Strategic Piles of Size k

Permutations with strategic pile $\{b_1, \ldots, b_k\}$ take the form:

$$[b_k + 1 \ \ldots \ b_{x_1} \ \underbrace{b_{x_1 - 1} + 1 \ \ldots \ b_{x_2}}_{\text{merge if equal}} \ b_{x_2 - 1} + 1 \ \cdots \ \underbrace{b_{x_{k-1}} \ b_{x_{k-1} - 1} + 1}_{\text{pair}} \ \ldots \ b_1].$$

Form for Strategic Piles of Size k

Permutations with strategic pile $\{b_1, \ldots, b_k\}$ take the form:

$$[b_k + 1 \ \ldots \ b_{x_1} \ \underbrace{b_{x_1 - 1} + 1 \ \ldots \ b_{x_2}}_{\text{merge if equal}} \ b_{x_2 - 1} + 1 \ \cdots \ \underbrace{b_{x_{k-1}} \ b_{x_{k-1} - 1} + 1}_{\text{pair}} \ \ldots \ b_1].$$

Together, the entries b_{x_i} $b_{x_i - 1} + 1$ are called a **pair**. A **merge** occurs when $b_{x_i - 1} + 1 = b_{x_{i+1}}$.

Example

Consider the permutation $\pi = [5\ 2\ 4\ 6\ 3\ 7\ 1]$ with $SP(\pi) = \{1, 5, 4\}$.

Example

Consider the permutation $\pi = [5\ 2\ 4\ 6\ 3\ 7\ 1]$ with $SP(\pi) = \{1, 5, 4\}$.

$$\begin{array}{lccccccccl} [& 5 & 2 & 4 & 6 & 3 & 7 & 1 &] \\ [& b_3 + 1 & b_1 + 1 & b_3 & b_2 + 1 & 3 & 7 & b_1 &] \\ & b_2 & & & & & & & \end{array}$$

Example

Consider the permutation $\pi = [5\ 2\ 4\ 6\ 3\ 7\ 1]$ with $SP(\pi) = \{1, 5, 4\}$.

$$\begin{array}{lcccccccl} [& 5 & 2 & 4 & 6 & 3 & 7 & 1 &] \\ [& b_3 + 1 & b_1 + 1 & b_3 & b_2 + 1 & 3 & 7 & b_1 &] \\ & b_2 & & & & & & & \end{array}$$

Because $b_3 + 1 = b_2$, there is one merge in this permutation.

Patterns

k	**Number of permutations in S_n with strategic pile of size k**

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!$
2	$(n-3)(n-2)!$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!$
2	$(n-3)(n-2)!$
3	$(n-3)!\,\left[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\,\right]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!$
2	$(n-3)(n-2)!$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\,]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}16 + \binom{n-5}{1}16\,]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\left[\binom{n-2}{0}0!\ \right]$
2	$(n-2)!\left[\binom{n-3}{1}1!\ \right]$
3	$(n-3)!\left[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\right]$
4	$(n-4)!\left[\binom{n-5}{3}3! + \binom{n-5}{2}16 + \binom{n-5}{1}16\right]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}16 + \binom{n-5}{1}16\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4! + \binom{n-6}{3}\_\_\_ + \binom{n-6}{2}\_\_\_ + \binom{n-6}{1}\_\_\_ + \binom{n-6}{0}\_\_\_\ ]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}16 + \binom{n-5}{1}16\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4! + \binom{n-6}{3}90 + \binom{n-6}{2}130 + \binom{n-6}{1}80 + \binom{n-6}{0}90\ ]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}3 + \binom{n-4}{0}3\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}16 + \binom{n-5}{1}16\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4! + \binom{n-6}{3}90 + \binom{n-6}{2}130 + \binom{n-6}{1}80 + \binom{n-6}{0}90\ ]$
6	$(n-6)!\,[\binom{n-7}{5}5! + \binom{n-7}{4}\underline{\quad\quad} + \binom{n-7}{3}\underline{\quad\quad} + \binom{n-7}{2}\underline{\quad\quad} + \binom{n-7}{1}\underline{\quad\quad}\ ]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2!+\binom{n-4}{1}3+\binom{n-4}{0}3\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3!+\binom{n-5}{2}16+\binom{n-5}{1}16\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4!+\binom{n-6}{3}90+\binom{n-6}{2}130+\binom{n-6}{1}80+\binom{n-6}{0}90\ ]$
6	$(n-6)!\,[\binom{n-7}{5}5!+\binom{n-7}{4}576+\binom{n-7}{3}1116+\binom{n-7}{2}1080+\binom{n-7}{1}540\ ]$

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}c_{3,1} + \binom{n-4}{0}c_{3,2}\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}c_{4,1} + \binom{n-5}{1}c_{4,2}\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4! + \binom{n-6}{3}c_{5,1} + \binom{n-6}{2}c_{5,2} + \binom{n-6}{1}c_{5,3} + \binom{n-6}{0}c_{5,4}\ ]$
6	$(n-6)!\,[\binom{n-7}{5}5! + \binom{n-7}{4}c_{6,1} + \binom{n-7}{3}c_{6,2} + \binom{n-7}{2}c_{6,3} + \binom{n-7}{1}c_{6,4}\ ]$

where $c_{k,l}$ is called a **merge number** and represents the number of ways $k-1$ pairs can be ordered and have l merges

Patterns

k	**Number of permutations in S_n with strategic pile of size k**
1	$(n-1)!\,[\binom{n-2}{0}0!\ ]$
2	$(n-2)!\,[\binom{n-3}{1}1!\ ]$
3	$(n-3)!\,[\binom{n-4}{2}2! + \binom{n-4}{1}c_{3,1} + \binom{n-4}{0}c_{3,2}\ ]$
4	$(n-4)!\,[\binom{n-5}{3}3! + \binom{n-5}{2}c_{4,1} + \binom{n-5}{1}c_{4,2}\ ]$
5	$(n-5)!\,[\binom{n-6}{4}4! + \binom{n-6}{3}c_{5,1} + \binom{n-6}{2}c_{5,2} + \binom{n-6}{1}c_{5,3} + \binom{n-6}{0}c_{5,4}\ ]$
6	$(n-6)!\,[\binom{n-7}{5}5! + \binom{n-7}{4}c_{6,1} + \binom{n-7}{3}c_{6,2} + \binom{n-7}{2}c_{6,3} + \binom{n-7}{1}c_{6,4}\ ]$
k	$(n-k)!\,[\binom{n-(k+1)}{k-1}(k-1)! + \binom{n-(k+1)}{k-2}c_{k,1} + \cdots + \binom{n-(k+1)}{1}c_{k,k-2}\ ]$

where $c_{k,l}$ is called a **merge number** and represents the number of ways $k-1$ pairs can be ordered and have l merges

Strategic Piles of Size k

Theorem

The number of permutations of length n with strategic pile size k is

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

for k odd, and

$$(n-k)!\sum_{i=0}^{k-2} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

for k even.

Strategic Piles of Size k

Formula: k odd

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

Strategic Piles of Size k

Formula: k odd

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

- There are $(n-k)!$ ways to assign $\{1,\ldots,n\}$ to $\{b_1,\ldots,b_k\}$ and order the remaining elements.

Strategic Piles of Size k

Formula: k odd

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

- There are $(n-k)!$ ways to assign $\{1,\ldots,n\}$ to $\{b_1,\ldots,b_k\}$ and order the remaining elements.
- The i^{th} term represents arranging pairs using i merges.

Strategic Piles of Size k

Formula: k odd

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

- There are $(n-k)!$ ways to assign $\{1,\ldots,n\}$ to $\{b_1,\ldots,b_k\}$ and order the remaining elements.
- The i^{th} term represents arranging pairs using i merges.
  - Each $c_{k,i}$ counts the ways to order pairs using i merges.

Strategic Piles of Size k

Formula: k odd

$$(n-k)!\sum_{i=0}^{k-1} c_{k,i}\binom{n-(k+1)}{k-(i+1)}$$

- There are $(n-k)!$ ways to assign $\{1,\ldots,n\}$ to $\{b_1,\ldots,b_k\}$ and order the remaining elements.
- The i^{th} term represents arranging pairs using i merges.
  - Each $c_{k,i}$ counts the ways to order pairs using i merges.
  - The binomial coefficients count the ways to place these ordered pairs.

Merge Numbers

Merge Number Definition

Merge Number Definition

Definition

A **merge number**, $c_{k,l}$, is the number of ways to order the k pairs and choose l merges.

Merge Number Definition

Definition

A **merge number**, $c_{k,l}$, is the number of ways to order the k pairs and choose l merges.

Example: Consider strategic piles of size 3. The merge number, $c_{3,1}$, represents the number of ways to order the 2 free pairs in a permutation which has one merge. There are only two orderings of the pairs for $k = 3$:

Merge Number Definition

Definition

A **merge number**, $c_{k,l}$, is the number of ways to order the k pairs and choose l merges.

Example: Consider strategic piles of size 3. The merge number, $c_{3,1}$, represents the number of ways to order the 2 free pairs in a permutation which has one merge. There are only two orderings of the pairs for $k = 3$:

1. $[b_3 + 1, \ldots b_2, b_1 + 1, \ldots b_3, b_2 + 1, \ldots b_1]$

Merge Number Definition

Definition

A **merge number**, $c_{k,l}$, is the number of ways to order the k pairs and choose l merges.

Example: Consider strategic piles of size 3. The merge number, $c_{3,1}$, represents the number of ways to order the 2 free pairs in a permutation which has one merge. There are only two orderings of the pairs for $k = 3$:

1. $[b_3 + 1, \dots b_2, b_1 + 1, \dots b_3, b_2 + 1, \dots b_1]$
2. $[b_3 + 1, \dots b_3, b_2 + 1, \dots b_2, b_1 + 1, \dots b_1]$

Merge Number Definition

Definition

A **merge number**, $c_{k,l}$, is the number of ways to order the k pairs and choose l merges.

Example: Consider strategic piles of size 3. The merge number, $c_{3,1}$, represents the number of ways to order the 2 free pairs in a permutation which has one merge. There are only two orderings of the pairs for $k = 3$:

1. $[b_3 + 1, \ldots b_2, b_1 + 1, \ldots b_3, b_2 + 1, \ldots b_1]$
2. $[b_3 + 1, \ldots b_3, b_2 + 1, \ldots b_2, b_1 + 1, \ldots b_1]$

Thus we can count the number of places a merge can occur in each form. A merge can occur between any of the pairs in the first form, and between none and the second. Therefore $c_{3,1} = 3 + 0 = 3$.

Merge Graphs

Given a permutation, we can define a directed graph to represent the merges present in that permutation. Let each strategic pile element be a vertex. There is an edge from b_i to b_j if and only if $b_i + 1 = b_j$

Merge Graphs

For example, consider the following permutation with strategic pile $\{9, 7, 4, 3, 2\}$:

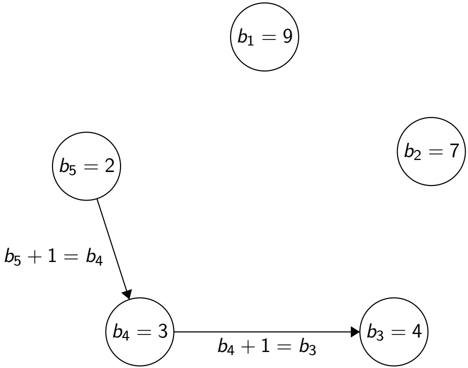
$$
\begin{array}{cccccccccccc}
[& 3 & 5 & 1 & 7 & 10 & 2 & 4 & 8 & 6 & 9 &] \\
 & b_4 & & & b_2 & & b_5 & b_3 & & & b_1 & \\
 & b_5+1 & b_3+1 & & & b_1+1 & & b_4+1 & b_2+1 & & &
\end{array}
$$

Merge Graphs

For example, consider the following permutation with strategic pile $\{9, 7, 4, 3, 2\}$:

[	3	5	1	7	10	2	4	8	6	9	]
	b_4			b_2		b_5	b_3			b_1	
	$b_5 + 1$	$b_3 + 1$			$b_1 + 1$		$b_4 + 1$	$b_2 + 1$			

The corresponding merge graph would be:

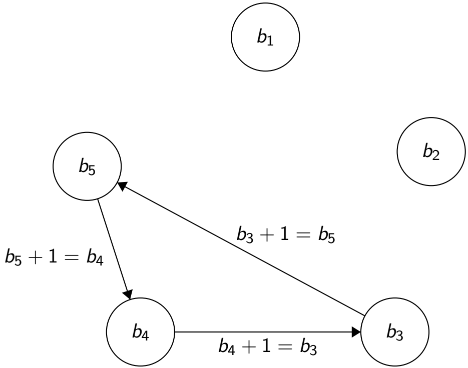


Merge Graphs

It is important to note that merge graphs cannot contain a cycle, otherwise we get a contradiction.

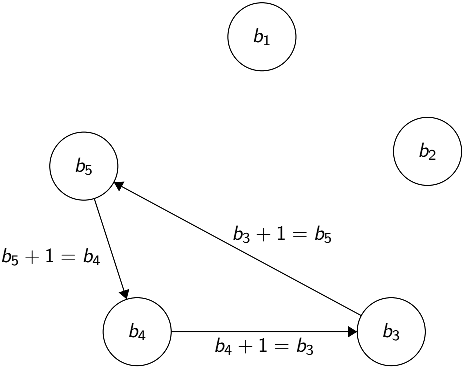
Merge Graphs

It is important to note that merge graphs cannot contain a cycle, otherwise we get a contradiction. For example, if we are given the following graph:



Merge Graphs

It is important to note that merge graphs cannot contain a cycle, otherwise we get a contradiction. For example, if we are given the following graph:



we know it cannot represent a merge graph. If all three of the equalities given are true, $b_5 + 1 = b_5 - 2$, a contradiction

Graphs of Possible Merges

Graphs of Possible Merges

- Given a pair ordering, we are interested in the merges that permutations with this ordering have

Graphs of Possible Merges

- Given a pair ordering, we are interested in the merges that permutations with this ordering have
- For example, if we are considering permutations with strategic pile of size 4, and we know that the strategic pile elements appear in the order

  $$[b_5{+}1, \ldots, b_5, b_4{+}1, \ldots, b_3, b_2{+}1, \ldots, b_2, b_1{+}1, \ldots, b_4, b_3{+}1, \ldots, b_1]$$

  we know that it's possible $b_4 + 1 = b_3$, but that $b_4 + 1$ cannot equal b_2.

Graphs of Possible Merges

- Given a pair ordering, we are interested in the merges that permutations with this ordering have
- For example, if we are considering permutations with strategic pile of size 4, and we know that the strategic pile elements appear in the order

  $$[b_5{+}1, \ldots, b_5, b_4{+}1, \ldots, b_3, b_2{+}1, \ldots, b_2, b_1{+}1, \ldots, b_4, b_3{+}1, \ldots, b_1]$$

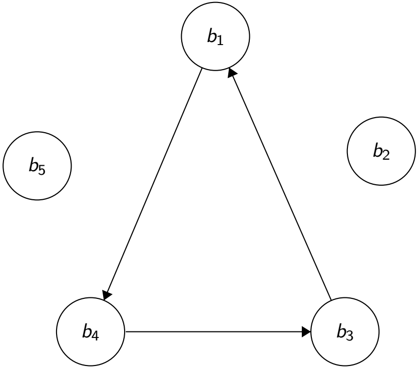
  we know that it's possible $b_4 + 1 = b_3$, but that $b_4 + 1$ cannot equal b_2.
- We represent these possible merges in a graph with a vertex for each strategic pile element. We draw an edge from b_i to b_j if and only if, given the pair ordering, $b_i + 1 = b_j$ is a possible merge. This is equivalent to saying that $b_i + 1$ appears just before b_j in the pair ordering.

Graph of Possible Merges - Example

$$[b_5+1,\ldots,b_5,b_4+1,\ldots,b_3,b_2+1,\ldots,b_2,b_1+1,\ldots,b_4,b_3+1,\ldots,b_1]$$

Graph of Possible Merges - Example

$$[b_5+1,\ldots,b_5,b_4+1,\ldots,b_3,b_2+1,\ldots,b_2,b_1+1,\ldots,b_4,b_3+1,\ldots,b_1]$$



Graph of Possible Merges

Some properties of this graph include:

Graph of Possible Merges

Some properties of this graph include:

- The graph is always composed of disjoint cycles and isolated points.

Graph of Possible Merges

Some properties of this graph include:

- The graph is always composed of disjoint cycles and isolated points.
- The map from a pair ordering to its graph of possible merges is injective.

Graph of Possible Merges

Some properties of this graph include:

- The graph is always composed of disjoint cycles and isolated points.
- The map from a pair ordering to its graph of possible merges is injective.
- The merge graph for a permutation is an edge subgraph of the graph of possible merges for its pair ordering.

Proposed Algorithm

The following process can be used to find $c_{k,l}$:

Proposed Algorithm

The following process can be used to find $c_{k,l}$:

1. Determine all possible cycle structures that a Possible Merge Graph with k vertices can have. This is equivalent to looking at all partitions of every integer $a \leq k$ such that the partitions have no parts of size 1 as well as an even number of even parts.

Proposed Algorithm

The following process can be used to find $c_{k,l}$:

1. Determine all possible cycle structures that a Possible Merge Graph with k vertices can have. This is equivalent to looking at all partitions of every integer $a \leq k$ such that the partitions have no parts of size 1 as well as an even number of even parts.
2. For each of these partitions:

Proposed Algorithm

The following process can be used to find $c_{k,l}$:

1. Determine all possible cycle structures that a Possible Merge Graph with k vertices can have. This is equivalent to looking at all partitions of every integer $a \leq k$ such that the partitions have no parts of size 1 as well as an even number of even parts.
2. For each of these partitions:
    1. Determine how many free pair orderings for strategic pile size k have the given cycle structure

Proposed Algorithm

The following process can be used to find $c_{k,l}$:

1. Determine all possible cycle structures that a Possible Merge Graph with k vertices can have. This is equivalent to looking at all partitions of every integer $a \leq k$ such that the partitions have no parts of size 1 as well as an even number of even parts.
2. For each of these partitions:
    1. Determine how many free pair orderings for strategic pile size k have the given cycle structure
    2. Use the inclusion-exclusion formula to figure out how many ways l merges can be picked with the given cycle structure

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4, 2]$, $[3, 3]$, $[5]$, $[2, 2]$, and $[3]$.

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4,2]$, $[3,3]$, $[5]$, $[2,2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4,2]$

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4,2]$, $[3,3]$, $[5]$, $[2,2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4,2]$
   1. There are 12 graphs with 6 vertices and the cycles structure $[4,2]$.

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4,2]$, $[3,3]$, $[5]$, $[2,2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4,2]$
    1. There are 12 graphs with 6 vertices and the cycles structure $[4,2]$.
    2. We must choose 3 edges from the 6 edges given in the cycle structure, but we cannot choose a complete cycle

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4,2]$, $[3,3]$, $[5]$, $[2,2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4,2]$
   1. There are 12 graphs with 6 vertices and the cycles structure $[4,2]$.
   2. We must choose 3 edges from the 6 edges given in the cycle structure, but we cannot choose a complete cycle
   3. In other words, we can choose any combination of the three edges, except for a combination that includes both edges of the 2-cycle. There are $\binom{6}{3} - 4$ ways to do this.

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4, 2]$, $[3, 3]$, $[5]$, $[2, 2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4, 2]$
   1. There are 12 graphs with 6 vertices and the cycles structure $[4, 2]$.
   2. We must choose 3 edges from the 6 edges given in the cycle structure, but we cannot choose a complete cycle
   3. In other words, we can choose any combination of the three edges, except for a combination that includes both edges of the 2-cycle. There are $\binom{6}{3} - 4$ ways to do this.
   4. For general cases, this process can be carried out using Inclusion-Exclusion.

Proposed Algorithm - Example

Assume we are trying to find $c_{6,3}$:

1. We first determine that the valid cycle structures are $[4,2]$, $[3,3]$, $[5]$, $[2,2]$, and $[3]$.
2. We first consider graphs with a cycle structure of $[4,2]$
    1. There are 12 graphs with 6 vertices and the cycles structure $[4,2]$.
    2. We must choose 3 edges from the 6 edges given in the cycle structure, but we cannot choose a complete cycle
    3. In other words, we can choose any combination of the three edges, except for a combination that includes both edges of the 2-cycle. There are $\binom{6}{3} - 4$ ways to do this.
    4. For general cases, this process can be carried out using Inclusion-Exclusion.
    5. Therefore, there are a total of $12[\binom{6}{3} - 4]$ free pair orderings with the given cycle structure

Proposed Algorithm - Example

Proposed Algorithm - Example

3. We proceed similarly for each of the other listed cycle structures

Proposed Algorithm - Example

3. We proceed similarly for each of the other listed cycle structures
4. We sum the results for each cycle structure to get $c_{6,3}$:

$$c_{6,3} = \underbrace{24\left[\binom{6}{3} - 4\right]}_{[4,2]} + \underbrace{12\left[\binom{6}{3} - 2\right]}_{[3,3]} + \underbrace{48\left[\binom{5}{3}\right]}_{[5]} + \underbrace{15\left[0\right]}_{[2,2]} + \underbrace{20\left[0\right]}_{[3,3]} = 1080$$

Proposed Algorithm - Analysis: Step 1

Proposed Algorithm - Analysis: Step 1

- The first step of the algorithm involves finding partitions of integers that meet certain criteria

Proposed Algorithm - Analysis: Step 1

- The first step of the algorithm involves finding partitions of integers that meet certain criteria
- Brute force algorithms to do this are $O(n^n)$, however, a faster algorithm may exist.

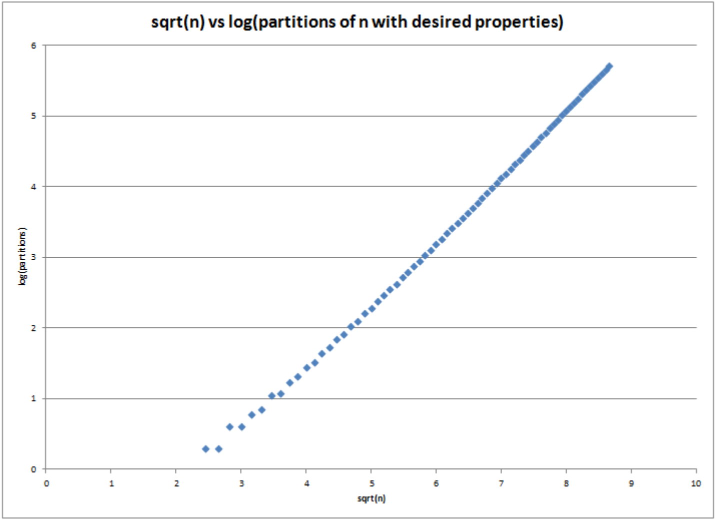
Proposed Algorithm - Analysis: Step 1

- The first step of the algorithm involves finding partitions of integers that meet certain criteria
- Brute force algorithms to do this are $O(n^n)$, however, a faster algorithm may exist.
- Data suggests that, given an integer m, the number of partitions we are interested in is $\Theta(e^{\sqrt{m}})$.

Proposed Algorithm - Analysis: Step 1

- The first step of the algorithm involves finding partitions of integers that meet certain criteria
- Brute force algorithms to do this are $O(n^n)$, however, a faster algorithm may exist.
- Data suggests that, given an integer m, the number of partitions we are interested in is $\Theta(e^{\sqrt{m}})$.
- Assuming this is true, this step of the algorithm is no faster than $\Omega(e^{\sqrt{k}})$.

Proposed Algorithm - Analysis: Step 1



Proposed Algorithm - Analysis: Step 2

Proposed Algorithm - Analysis: Step 2

- The first part of step two involves determining how many pair orderings have a given cycle structure in their possible merge graph.

Proposed Algorithm - Analysis: Step 2

- The first part of step two involves determining how many pair orderings have a given cycle structure in their possible merge graph.
- We have a recursive formula for determining this; however it is incomplete since we do not know how to find the base case for this formula.

Step 2 - Recurrence Relation

Theorem

Let $e_{k,\{a_1,a_2,\ldots,a_i\}}$ *be the number of orderings with a strategic pile of size* k *and Graph of Possible Merges with the given cycle structure. Then*

$$e_{k,[a_1,a_2,\ldots,a_i]} = \frac{e_{k-1,[a_1,a_2,\ldots,a_i]}k}{k-(a_1+a_2+\ldots+a_i)}$$

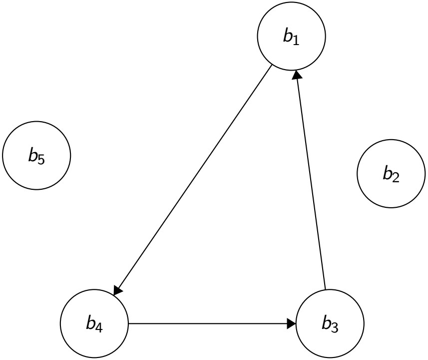
Step 2 - Recurrence Relation: Proof Sketch

Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:

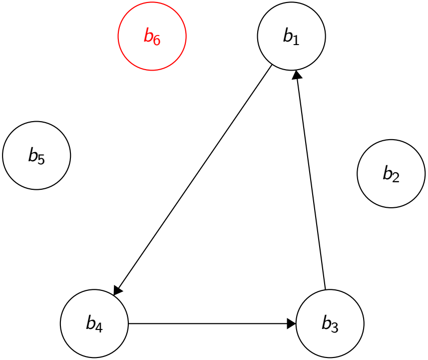
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



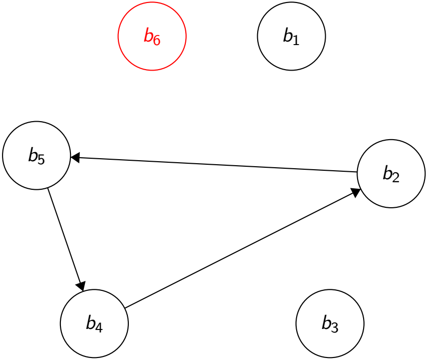
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



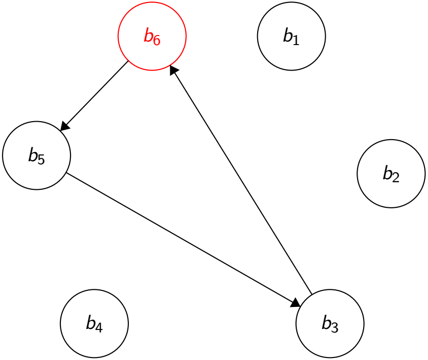
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



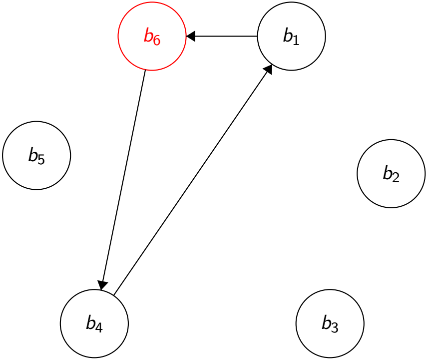
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



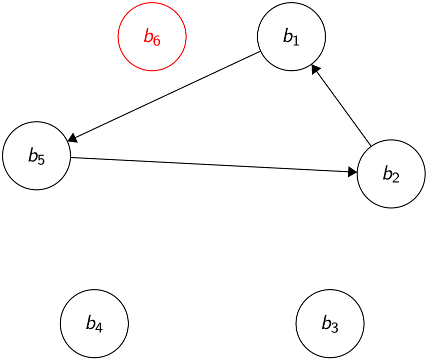
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



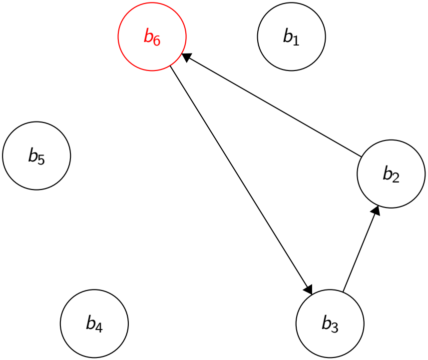
Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



Step 2 - Recurrence Relation: Proof Sketch

- The set of Possible Merge Graphs with k vertices that come from pair orderings is exactly the set of of graphs formed in the following manner:



Step 2 - Recurrence Relation: Proof Sketch

- Since each graph can be rotated k times, we multiply k by $e_{k,[a_1,a_2,....,a_i]}$ to get the order of this set.

Step 2 - Recurrence Relation: Proof Sketch

- Since each graph can be rotated k times, we multiply k by $e_{k,[a_1,a_2,\ldots,a_i]}$ to get the order of this set.
- However, we have counted some graphs multiple times. Using the orbit-stabilizer theorem, we can deduce that we have counted each Possible Merge Graph $k-(a_1+a_2+\ldots+a_i)$ times, giving us the given formula.

Step 2 - Recurrence Relation: Proof Sketch

- Since each graph can be rotated k times, we multiply k by $e_{k,[a_1,a_2,....,a_i]}$ to get the order of this set.
- However, we have counted some graphs multiple times. Using the orbit-stabilizer theorem, we can deduce that we have counted each Possible Merge Graph $k - (a_1 + a_2 + ... + a_i)$ times, giving us the given formula.
- Using the recursive formula to find the number of pair orderings that have a given cycle structure runs in no less than polynomial time, as it requires $k - l$ recursive steps.

Proposed Algorithm - Analysis: Step 2

Proposed Algorithm - Analysis: Step 2

- The second part of step two involves using inclusion exclusion to find the number of ways l merges can be picked assuming the given cycle structure.

Proposed Algorithm - Analysis: Step 2

- The second part of step two involves using inclusion exclusion to find the number of ways l merges can be picked assuming the given cycle structure.
- The number of ways l edges that contain a cycle can be picked is:

$$\sum_{j=1}^{i}(-1)^{j+1}\sum\left\{\binom{e-a_{k_1}-a_{k_2}-\cdots-a_{k_j}}{l-a_{k_1}-a_{k_2}-\cdots-a_{k_j}} : 1 \leq k_1, k_2, ..., k_j \leq i, \mathit{all}\ \ k\ \ \mathit{distinct}\right\}$$

Proposed Algorithm - Analysis: Step 2

- The second part of step two involves using inclusion exclusion to find the number of ways l merges can be picked assuming the given cycle structure.
- The number of ways l edges that contain a cycle can be picked is:

$$\sum_{j=1}^{i}(-1)^{j+1}\sum\left\{\binom{e-a_{k_1}-a_{k_2}-\cdots-a_{k_j}}{l-a_{k_1}-a_{k_2}-\cdots-a_{k_j}} : 1\leq k_1,k_2,...,k_j\leq i, \mathit{all}\ k\ \mathit{distinct}\right\}$$

- For a given cycle structure, calculating this will require 2^i iterations

Proposed Algorithm - Analysis: Step 2

- The second part of step two involves using inclusion exclusion to find the number of ways l merges can be picked assuming the given cycle structure.
- The number of ways l edges that contain a cycle can be picked is:

$$\sum_{j=1}^{i}(-1)^{j+1}\sum\left\{\binom{e-a_{k_1}-a_{k_2}-\cdots-a_{k_j}}{l-a_{k_1}-a_{k_2}-\cdots-a_{k_j}} : 1 \leq k_1, k_2, ..., k_j \leq i, \textit{all } k \textit{ distinct}\right\}$$

- For a given cycle structure, calculating this will require 2^i iterations
- Thus, for each cycle structure, this part of the algorithm can be done in $\Theta(e^k)$ steps.

Proposed Algorithm - Analysis: Step 2

- The second part of step two involves using inclusion exclusion to find the number of ways l merges can be picked assuming the given cycle structure.
- The number of ways l edges that contain a cycle can be picked is:

$$\sum_{j=1}^{i}(-1)^{j+1}\sum\left\{\binom{e-a_{k_1}-a_{k_2}-\cdots-a_{k_j}}{l-a_{k_1}-a_{k_2}-\cdots-a_{k_j}} : 1 \leq k_1, k_2, ..., k_j \leq i, \mathit{all}\ \ k\ \ \mathit{distinct}\right\}$$

- For a given cycle structure, calculating this will require 2^i iterations
- Thus, for each cycle structure, this part of the algorithm can be done in $\Theta(e^k)$ steps.
- Since the first part of step 2 can be done in no less than polynomial time, step 2 in it's entirety should take $\Omega(e^k)$ steps for each cycle structure

Propose Algorithm - Analysis

Propose Algorithm - Analysis

- The running time of Step 1 is $\Omega(e^{\sqrt{k}})$, assuming our conjecture on the number of cycle structures in the desired form is true

Propose Algorithm - Analysis

- The running time of Step 1 is $\Omega(e^{\sqrt{k}})$, assuming our conjecture on the number of cycle structures in the desired form is true
- Step 2 will take $\Omega(e^k)$ steps for each cycle structure

Propose Algorithm - Analysis

- The running time of Step 1 is $\Omega(e^{\sqrt{k}})$, assuming our conjecture on the number of cycle structures in the desired form is true
- Step 2 will take $\Omega(e^k)$ steps for each cycle structure
- Assuming there are $\Theta(e^{\sqrt{k}})$ cycle structures, our algorithm can be no faster than $\Omega(e^{k^{3/2}})$

Future Work

Future Work

Future Work

- Complete our merge number algorithm, which would require:

Future Work

- Complete our merge number algorithm, which would require:
  - Determining the integer partitions of n with all parts of size greater than one and an even number of even parts.

Future Work

- Complete our merge number algorithm, which would require:
  - Determining the integer partitions of n with all parts of size greater than one and an even number of even parts.
  - Establishing the base case for a recursive formula.

Future Work

- Complete our merge number algorithm, which would require:
  - Determining the integer partitions of n with all parts of size greater than one and an even number of even parts.
  - Establishing the base case for a recursive formula.
- Construct an alternative merge number algorithm that will run in less than exponential time.

References

- K.L.M. Adamyk, E. Holmes, G.R. Mayfield, D.J. Moritz, M. Scheepers, B.E. Tenner and H.C. Wauck, *Sorting Permutations: Games, Genomes and Cycles*, **arXiv:1410.2353**
- L. Cangelmi, *Factorization of an n-cycle into two n-cycles*, **European Journal of Combinatorics** 24 (2003) 849–853.
- D.M. Prescott, A. Ehrenfeucht and G. Rozenberg, *Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates*, **Journal of Theoretical Biology** 222 (2003), 323 - 330.

Acknowledgments

Thanks to the National Science Foundation, Grant No. DMS-1359425, and to Boise State University.