Shuffle Sorting Permutations

Lara Pudwell[1] Rebecca Smith[2]

[1]Valparaiso University

[2]SUNY Brockport

Day of Lara's Birth, 2022

The Plan:

Introduction

The Plan:

The Plan:

The Plan:

The Plan:

Introduction

Definition

An *ascent* of permutation π is an index i where $\pi_i < \pi_{i+1}$, while a *descent* is an index i where $\pi_i > \pi_{i+1}$.

Definition

An *ascent* of permutation π is an index i where $\pi_i < \pi_{i+1}$, while a *descent* is an index i where $\pi_i > \pi_{i+1}$.

We denote the number of ascents of π by $\mathrm{asc}(\pi)$ and the number of descents by $\mathrm{des}(\pi)$.

Definition

An *ascent* of permutation π is an index i where $\pi_i < \pi_{i+1}$, while a *descent* is an index i where $\pi_i > \pi_{i+1}$.

We denote the number of ascents of π by $\mathrm{asc}(\pi)$ and the number of descents by $\mathrm{des}(\pi)$.

Example

Given the permutation $\pi = 35841726$, notice $\mathrm{asc}(\pi) = 4$ and $\mathrm{des}(\pi) = 3$.

Definition

A *sorting function* is a function $f : \mathcal{S}_n \to \mathcal{S}_n$ such that for all $\pi \in \mathcal{S}_n$, there exists $i \in \mathbb{N}$ such that $f^{\,i}(\pi) = 123 \cdots n$.

Definition

A *sorting function* is a function $f : \mathcal{S}_n \to \mathcal{S}_n$ such that for all $\pi \in \mathcal{S}_n$, there exists $i \in \mathbb{N}$ such that $f^{\,i}(\pi) = 123 \cdots n$.

We develop four sorting functions motivated by shuffling cards.

In particular, a common way to shuffle is to cut a deck into two non-empty parts and then to *riffle* the two parts together, so that each part remains in order, but the two parts are interleaved.

To create shuffling algorithms that are both well-defined and sorting functions, we

- Determine where the cut is made and
- Create rules on how to riffle the parts.

To create shuffling algorithms that are both well-defined and sorting functions, we

- Determine where the cut is made and
- Create rules on how to riffle the parts.

In all four algorithms, the cut will be made immediately following the longest increasing prefix of the permutation.

To create shuffling algorithms that are both well-defined and sorting functions, we

- Determine where the cut is made and
- Create rules on how to riffle the parts.

In all four algorithms, the cut will be made immediately following the longest increasing prefix of the permutation.

Notation

Given a permutation π *with first descent at* $\pi_{i-1} > \pi_i$*, let* $\pi' = \pi_1 \cdots \pi_{i-1}$ *and* $\pi'' = \pi_i \cdots \pi_n$*.*

To create shuffling algorithms that are both well-defined and sorting functions, we

- Determine where the cut is made and
- Create rules on how to riffle the parts.

In all four algorithms, the cut will be made immediately following the longest increasing prefix of the permutation.

Notation

Given a permutation π with first descent at $\pi_{i-1} > \pi_i$, let $\pi' = \pi_1 \cdots \pi_{i-1}$ and $\pi'' = \pi_i \cdots \pi_n$.

Example

For the permutation $\pi = 35841726$, we have $\pi' = 358$ and $\pi'' = 41726$.

Definition

A *stack* is a last-in, first-out data structure with push and pop operations.

A *queue* is a first-in, first-out data structure.

Definition

A *stack* is a last-in, first-out data structure with push and pop operations.
A *queue* is a first-in, first-out data structure.

When we cut a deck and riffle it together, we may view this as a system of two queues.

Definition

A *stack* is a last-in, first-out data structure with push and pop operations.
A *queue* is a first-in, first-out data structure.

When we cut a deck and riffle it together, we may view this as a system of two queues.

When we cut a deck, then reverse the second half before riffling, this acts a system of a queue and a stack.

The PRE Algorithm

Prefix-preserving Shuffle: **PRE**

Given a permutation $\pi = \pi'\pi''$, the algorithm PRE acts according to the following rules:

1. If the next available entry b of π'' is smaller than the next available entry a of π' but larger than the current last entry of output (or $b < a$ and output is currently empty), then pop b to the output.
2. Else, if π' and π'' both still have entries, pop a to the output.
3. Once one of π' and π'' is empty, pop the remaining entries of the other sequence to the output.

Proposition

Any permutation π is sorted after exactly $\mathrm{des}(\pi)$ iterations of algorithm PRE.

Idea of Proof:

- The descent between π_{i-1} and π_i is removed.
- Any further descents must be contained entirely in π'', but these entries cannot be output until the end.
- No new descents will be introduced.

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow 358$

$\leftarrow 41726$

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow \quad 58$$

$$\leftarrow 41726$$

3

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \quad 58}}$$

$$\overline{\underline{\leftarrow \quad 1726}}$$

34

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$	8

$\leftarrow$	1726

345

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$

$\leftarrow \quad 1726$

3458

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\begin{array}{|l|}\hline \leftarrow \qquad \\ \hline \end{array}$$

$$\begin{array}{|lr|}\hline \leftarrow & 726 \\ \hline \end{array}$$

34581

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \qquad\quad}}$$

$$\overline{\underline{\leftarrow \qquad 26}}$$

345817

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$

$\leftarrow \quad 6$

3458172

Example

Apply PRE to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$

$\leftarrow$

34581726

The MIN Algorithm

Minimum-first Shuffle: **MIN**

Given a permutation $\pi = \pi'\pi''$, the algorithm MIN acts according to the following rules:

1. If the next available entry *a* of π' is smaller than the next available entry *b* of π'', then pop *a* to the output.

2. Else, if π' and π'' both still have entries, pop *b* to the output.

3. Once one of π' and π'' is empty, pop the remaining entries of the other sequence to the output.

Proposition

Any permutation π is sorted after exactly $\mathrm{des}(\pi)$ iterations of algorithm MIN.

Idea of Proof:

- The descent between π_{i-1} and π_i is removed.
- Any further descents must be contained entirely in π''. And any second element of a descent in π'' will be output immediately following the entry before it in π''.
- No new descents will be introduced.

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow 358$

$\leftarrow 41726$

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \ \ 58}}$$

$$\overline{\underline{\leftarrow 41726}}$$

3

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \quad 58}}$$

$$\overline{\underline{\leftarrow \quad 1726}}$$

34

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \quad 58}}$$

$$\overline{\underline{\leftarrow \quad 726}}$$

341

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\begin{array}{|cc|}\hline \leftarrow & 8 \\ \hline\end{array}$$

$$\begin{array}{|cc|}\hline \leftarrow & 726 \\ \hline\end{array}$$

3415

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \quad 8}}$$

$$\overline{\underline{\leftarrow \quad 26}}$$

34157

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \quad 8}}$$

$$\overline{\underline{\leftarrow \qquad 6}}$$

341572

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow \quad 8$$

$$\leftarrow$$

3415726

Example

Apply MIN to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow$$

$$\leftarrow$$

34157268

Corollary

The number of permutations in $\mathcal{S}_n$ that are sortable after exactly k passes of algorithm PRE or k passes of MIN is given by the Eulerian numbers (OEIS A008292).

The PRE-REV Algorithm

Prefix-preserving Reverse Shuffle: **PRE-REV**

Given a permutation $\pi = \pi'\pi''$, the algorithm PRE-REV acts according to the following rules:

1. If the next available entry b of $(\pi'')^{rev}$ is smaller than the next available entry a of π' but larger than the current last entry of output (or $b < a$ and output is currently empty), then pop b to the output.

2. Else, if π' and $(\pi'')^{rev}$ both still have entries, pop a to the output.

3. Once one of π' and $(\pi'')^{rev}$ is empty, pop the remaining entries of the other sequence to the output.

Proposition

Algorithm PRE-REV is a sorting function.

Notation

Define prefix-suffix decomposition of π *as follows: Let* $\pi^{(1)} = \pi' = \pi_1 \cdots \pi_{i-1}$ *be the longest increasing prefix of* π *and let* $\pi^{rev(1)} = (\pi'')^{rev}$ *be the reversal of* $\pi_i \cdots \pi_n$*.*

If $\pi^{rev(1)}$ *is empty, then we are done.*

Otherwise, given $\pi^{(1)}, \ldots, \pi^{(\ell)}$*, set* $\pi^{(\ell+1)}$ *to be the longest increasing prefix of* $\pi^{rev(\ell)}$ *and recursively define* $\pi^{rev(\ell+1)}$ *to be the reversal of the remaining digits.*

Example

The permutation $\pi = 562793841$ has

$$\begin{aligned} \pi^{(1)} &= \pi' = 56 & \pi^{rev(1)} &= (\pi'')^{rev} = 1483972 \\ \pi^{(2)} &= 148 & \pi^{rev(2)} &= 2793 \\ \pi^{(3)} &= 279 & \pi^{rev(3)} &= 3 \\ \pi^{(4)} &= 3 \end{aligned}$$

Example

The permutation $\pi = 562793841$ has

$$\begin{aligned} \pi^{(1)} = \pi' = 56 \quad & \pi^{rev(1)} = (\pi'')^{rev} = 1483972 \\ \pi^{(2)} = 148 \quad & \pi^{rev(2)} = 2793 \\ \pi^{(3)} = 279 \quad & \pi^{rev(3)} = 3 \\ \pi^{(4)} = 3 \quad & \end{aligned}$$

Theorem

Consider $\pi \in \mathcal{S}_n$. If there are $k+1$ parts in the prefix-suffix decomposition of π, then algorithm PRE-REV requires k iterations to sort π.

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow 358}}$$

$$\overline{\underline{41726 \rightarrow}}$$

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow\ 58}}$$

$$\overline{\underline{41726 \rightarrow}}$$

$$3$$

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$ 8

41726 $\rightarrow$

35

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow \quad 8$

$4172 \quad \rightarrow$

356

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overleftarrow{}$$

$$\overline{\underline{4172 \quad \rightarrow}}$$

3568

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$

417 $\rightarrow$

35682

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\begin{array}{|l|}\hline \leftarrow \\ \hline \end{array}$$

$$\begin{array}{|ll|}\hline 41 & \rightarrow \\ \hline \end{array}$$

356827

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow \qquad}}$$

$$\overline{\underline{4 \qquad \rightarrow}}$$

3568271

Example

Apply PRE-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow$$

$$\rightarrow$$

35682714

$n \backslash k$	0	1	2	3	4	5	6	7
1	1							
2	1	1						
3	1	3	2					
4	1	7	13	3				
5	1	15	58	40	6			
6	1	31	221	325	132	10		
7	1	63	774	2086	1711	385	20	
8	1	127	2577	11655	16841	7931	1153	35

Table: The number of permutations of length n sortable by exactly k applications of PRE-REV for small n and k.

The MIN-REV Algorithm

Minimum-first Reverse Shuffle: **MIN-REV**

Given a permutation $\pi = \pi'\pi''$, the algorithm MIN acts according to the following rules:

1. If the next available entry *a* of π' is smaller than the next available entry *b* of $(\pi'')^{rev}$, then pop *a* to the output.

2. Else, if π' and $(\pi'')^{rev}$ both still have entries, pop *b* to the output.

3. Once one of π' and $(\pi'')^{rev}$ is empty, pop the remaining entries of the other sequence to the output.

Proposition

Algorithm MIN-REV is a sorting function.

Proposition

Algorithm MIN-REV is a sorting function.

Theorem

Consider a non-identity permutation $\pi \in \mathcal{S}_n$*. Suppose there are* d *descents before* n *and* a *ascents after* n*. Then* π *requires exactly* $\max(2d, 2a + 1)$ *applications of MIN-REV to be sorted.*

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\overline{\underline{\leftarrow 358}}$$

$$\overline{\underline{41726 \rightarrow}}$$

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow \quad 58$$

$$41726 \rightarrow$$

3

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$ 8

41726 $\rightarrow$

35

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$ 8

4172 $\rightarrow$

356

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\begin{array}{cc} \hline \leftarrow & 8 \\ \hline \end{array}$$

$$\begin{array}{cc} \hline 417 & \rightarrow \\ \hline \end{array}$$

3562

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$ 8

41 $\rightarrow$

35627

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow \quad 8$$

$$4 \quad \rightarrow$$

356271

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$\leftarrow$ 8

$\rightarrow$

3562714

Example

Apply MIN-REV to to $\pi = 35841726$.

We shuffle $\pi' = 358$ and $\pi'' = 41726$ to get:

$$\leftarrow$$

$$\rightarrow$$

$$\color{magenta}{35}\color{black}{62714}\color{magenta}{8}$$

$n \backslash k$	0	1	2	3	4	5	6	7	8	9	10	11
1	1											
2	1	1										
3	1	3	1	1								
4	1	7	7	7	1	1						
5	1	15	33	39	15	15	1	1				
6	1	31	131	211	141	141	31	31	1	1		
7	1	63	473	1123	1128	1148	488	488	63	63	1	1

Table: The number of permutations of length n sortable by exactly k applications of MIN-REV for small n and k.

Proposition

Both the PRE-REV-sortable permutations and the MIN-REV-sortable permutations are precisely the unimodal permutations, i.e. the $\{213, 312\}$*-avoiding permutations.*

Example

The permutation $\pi = 24816753$ requires five iterations of the MIN-REV algorithm to be sorted:

$$MIN - REV(\pi) = 23457618$$
$$MIN - REV^2(\pi) = 23457816$$
$$MIN - REV^3(\pi) = 23456178$$
$$MIN - REV^4(\pi) = 23456871$$
$$MIN - REV^5(\pi) = 12345678$$

However, as $\pi = 24816753$ only requires two applications of the PRE-REV algorithm.

$$PRE - REV(\pi) = 23457861$$
$$PRE - REV^2(\pi) = 12345678$$

Proposition

Every permutation sortable after two iterations of algorithm MIN-REV is also sortable after two iterations of algorithm PRE-REV.

Proposition

Every permutation sortable after two iterations of algorithm MIN-REV is also sortable after two iterations of algorithm PRE-REV.

Proof.

The permutations sortable only after exactly two iterations of algorithm MIN-REV have one descent before n and no ascents after n. That is, they consist of two increasing sequences followed by a decreasing sequence. The prefix-suffix decomposition of such permutations have exactly these three parts. □

Theorem

The permutations of length n that are sortable by after two iterations of algorithm PRE-REV, but not two iterations of MIN-REV are counted by $S(n,3)$.

Theorem

The permutations of length n that are sortable by after two iterations of algorithm PRE-REV, but not two iterations of MIN-REV are counted by $S(n, 3)$.

Proof.

The permutations sortable by PRE-REV, but not MIN-REV are those of the form $\pi = \pi_1 \cdots n \Big| \pi_i \cdots \pi_{j-1} \Big| \pi_j \cdots \pi_n$ where $\pi_1 \cdots \pi_{i-1} = n$ is the maximum length increasing prefix, $\pi_i \cdots \pi_{j-1}$ is a nonempty increasing subsequence, and $\pi_j \cdots \pi_n$ is a nonempty decreasing subsequence where π_j is a peak of p and each of these sequences is nonempty.

These three sequences of entries are in bijection with (unlabeled) sets containing the corresponding entries since n is always in the first sequence and of the other two, $\{\pi_j, \ldots, \pi_n\}$ contains the largest remaining entry. $\square$

Thank you!