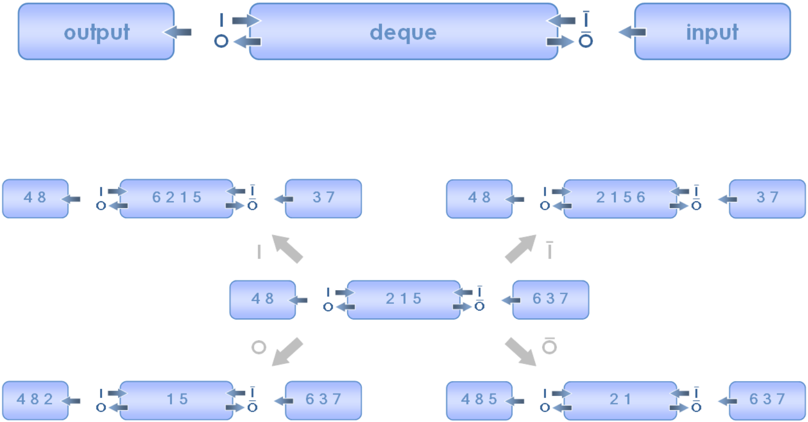
Deque sortable permutations and deterministic sorting procedures

Luca S. Ferrari

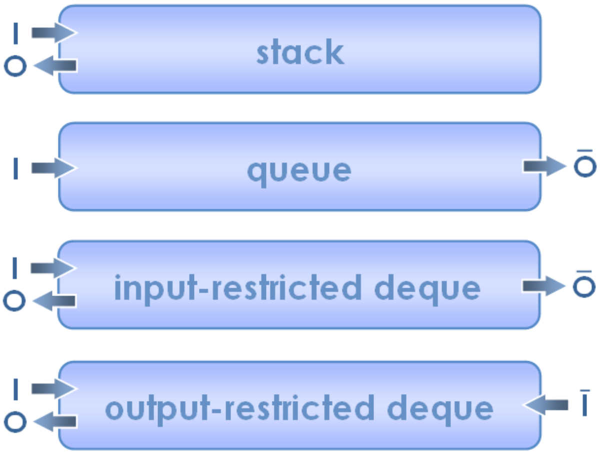
Dipartimento di Matematica
Università di Bologna

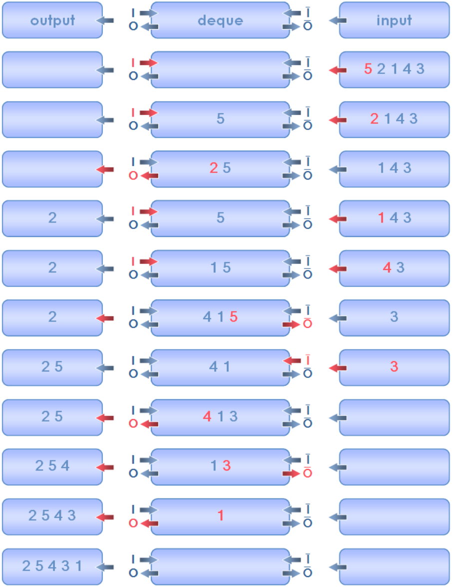
Permutation Patterns 2013
Paris

Data structures



Data structures



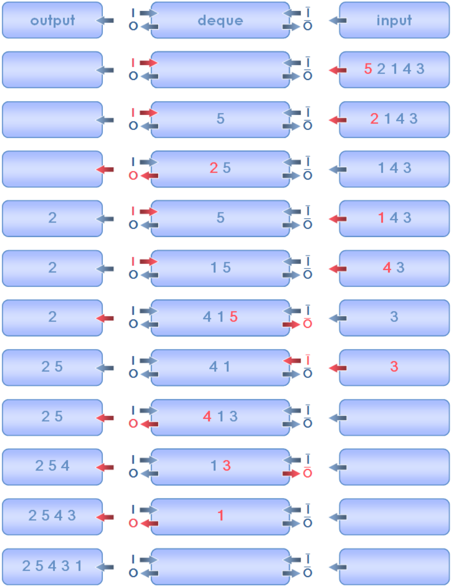


$$\sigma = 52143$$

$$S = IIOII\bar{O}\bar{I}O\bar{O}O$$

$$S(\sigma) = 25431$$

$\mathbb{X}$– sequences



$$\sigma = 52143$$

$$S = IIOII\bar{O}\bar{I}O\bar{O}O$$

$$S(\sigma) = 25431$$

$$id = 12345$$

$$S = IIOII\bar{O}\bar{I}O\bar{O}O$$

$$S(id) = 21453$$

$$S(\sigma) = \sigma \circ S(id) = 25431$$

Computable and sortable permutations

Definition

The permutation $S(id)$ is called the permutation *computed* by S.

Definition

We say that σ is *sorted* by S if $S(\sigma) = id$.

Proposition

A permutation σ is sorted by S if and only if its inverse σ^{-1} is computed by S:

$$S(\sigma) = id \iff \sigma^{-1} = S(id)$$

Sortable permutations

Definition

The set of permutations *sorted* by the device $\mathbb{X}$ is

$$Sort(\mathbb{X}) = \{\sigma \mid \exists S \in \mathcal{X} \mid S(\sigma) = id\}$$

Theorem (Knuth, 1968)

$$Sort(\mathbb{S}) = Av(231)$$

Sortable permutations

Theorem (Knuth, 1968 and West, 1995)

$$Sort(\mathbb{D}^{ir}) = Av(3241, 4231)$$

Theorem (Knuth, 1968 and West, 1995)

$$Sort(\mathbb{D}^{or}) = Av(2431, 4231)$$

Theorem (Pratt, 1973)

$$Sort(\mathbb{D}) = Av(T),$$

$$T = \{52341, 25341, 42351, 24351, 5274163, 2574163, 5264173, 2564173, \dots\}$$

$\mathbb{X}-$sorting sequences for σ

$$\mathcal{C}_{\mathbb{X}}(\sigma) = \{S \in \mathcal{X} \mid S(\sigma) = id\}$$

Example

$$\begin{aligned}
\mathcal{C}_{\mathbb{Q}}(312) &= \varnothing \\
\mathcal{C}_{\mathbb{S}}(312) &= \{IIOIOO\} \\
\mathcal{C}_{\mathbb{D}^{ir}}(312) &= \{IIOIOO, IIOIO\bar{O}\} \\
\mathcal{C}_{\mathbb{D}^{or}}(312) &= \{IIOIOO, \bar{I}IOIOO\} \\
\mathcal{C}_{\mathbb{D}}(312) &= \{IIOIOO, IIOIO\bar{O}, IIO\bar{I}\bar{O}O, IIO\bar{I}\bar{O}\bar{O}, I\bar{I}\bar{O}IOO, I\bar{I}\bar{O}IO\bar{O}, \\
&\quad I\bar{I}\bar{O}\bar{I}\bar{O}O, II\bar{I}O\bar{O}O, II\bar{I}O\bar{O}\bar{O}, I\bar{I}I\bar{O}OO, I\bar{I}I\bar{O}O\bar{O}, I\bar{I}\bar{O}\bar{I}\bar{O}\bar{O}, \\
&\quad \bar{I}I\bar{I}O\bar{O}O, \bar{I}I\bar{I}O\bar{O}\bar{O}, \bar{I}\bar{I}I\bar{O}OO, \bar{I}\bar{I}I\bar{O}O\bar{O}, \bar{I}IOIOO, \bar{I}IOIO\bar{O}, \\
&\quad \bar{I}IO\bar{I}\bar{O}O, \bar{I}IO\bar{I}\bar{O}\bar{O}, \bar{I}\bar{I}\bar{O}IOO, \bar{I}\bar{I}\bar{O}IO\bar{O}, \bar{I}\bar{I}\bar{O}\bar{I}\bar{O}O, \bar{I}\bar{I}\bar{O}\bar{I}\bar{O}\bar{O}\}
\end{aligned}$$

$\mathbb{X}-$Sorting Procedure

Problem

A permutation σ is sorted by a device $\mathbb{X}$?

Brute force approach | $\mathbb{X}-$Sorting Procedure

$\mathbb{X}-$Sorting Procedure

The $\mathbb{X}-$sorting procedure (X) must be:

- deterministic: finds $S_{\sigma,\mathbb{X}}$ without brute force approach;
- greedy: $Sort(X) = Sort(\mathbb{X})$;
- efficient: linear time complexity;
- general: applicable not only to permutations.

$\mathbb{X}$– Sorting Procedure

The procedure X

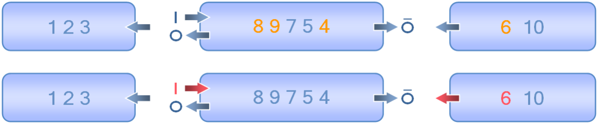
$$X:\quad Sort(\mathbb{X}) \longrightarrow \mathcal{X}$$
$$\sigma \longmapsto S_{\sigma,\mathbb{X}}.$$

$\mathbb{X}$– Sorting Procedure X: $\sigma \rightarrow \tau$

while $inside \neq \emptyset \vee input \neq \emptyset$ do

$S_{\sigma,\mathbb{X}}[step] \leftarrow \mathbb{X}\text{–}OperationChoice(inside[1,2,\ell], inp[1])$

end while

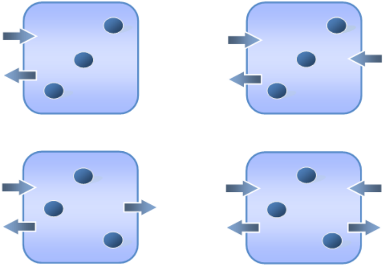


Monotonicity and unimodality

Proposition

Let $\mathbb{X}$ be the device used to sort an input permutation σ. Hence, at each state t of the sorting process:

- if $\mathbb{X} = \mathbb{S}$ or $\mathbb{X} = \mathbb{D}^{or}$, then *inside* is increasing;
- if $\mathbb{X} = \mathbb{D}^{ir}$ or $\mathbb{X} = \mathbb{D}$, then *inside* is unimodal.



Operation choice rules

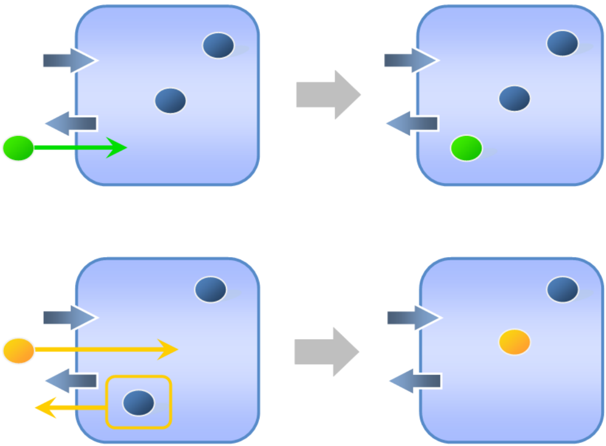
Rule 1

Always perform I or $\bar{I}$ instead of O or $\bar{O}$, whenever possible.

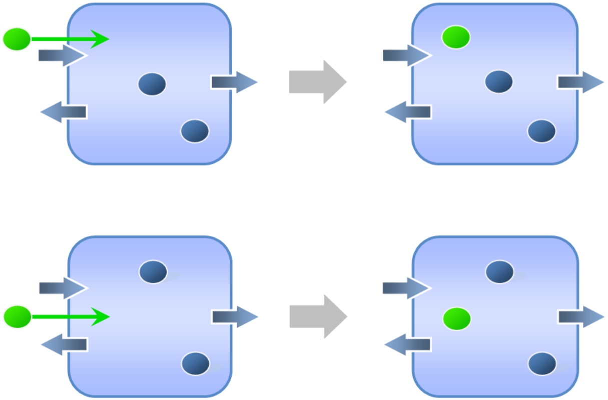
Rule 2

- If $inside = \varnothing$ and $input \neq \varnothing \Rightarrow$ Perform I
- If $inside = \{x\}$ and $input = \varnothing \Rightarrow$ Perform O

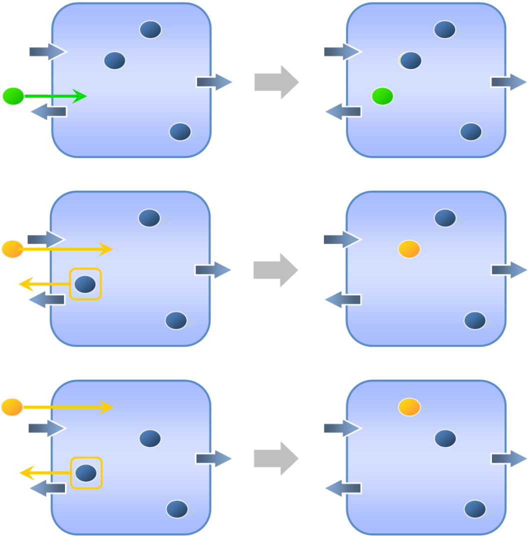
Stack



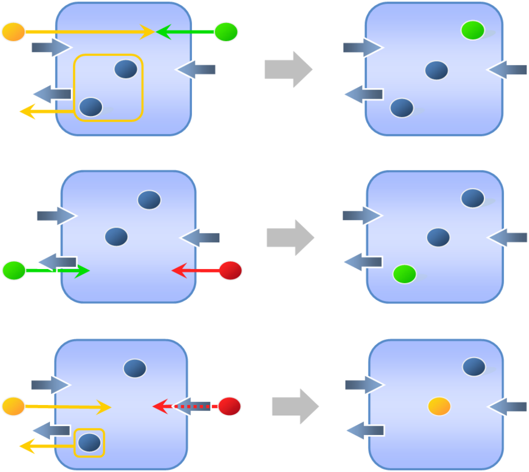
Input-restricted deque



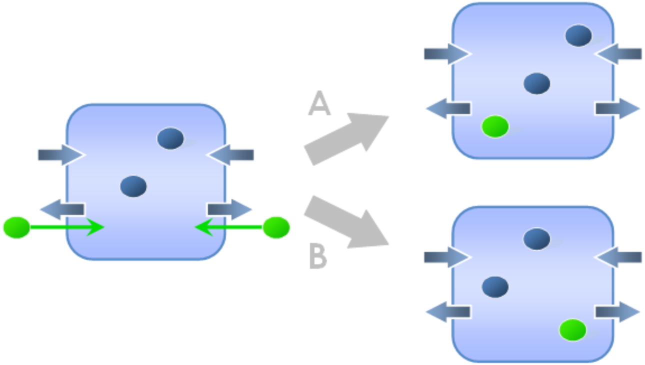
Input-restricted deque



Output-restricted deque



Deque



A fails with $\sigma_1 = 541362$ and B fails with $\sigma_2 = 43251$, although both σ_1 and σ_2 are deque sortable.

Sorting procedures and sorting algorithms

Proposition

$$X^{n-1}(\sigma) = id, \qquad \forall\, \sigma \in \Sigma_n$$

$\mathbb{X}$–Sorting Algorithm

$\mathbb{X}$–Sorting Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow X(\sigma)$

end for

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
  $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
  if $\sigma[j] > \sigma[j+1]$ then
    $swap\,(\sigma[j], \sigma[j+1])$
  end if
end for

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$3\,2\,5\,1\,4\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$\sigma = 3\,2\,5\,1\,4\,6$

$3\,2\,5\,1\,4\,6$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$\sigma = 3\,2\,5\,1\,4\,6$

$2\,3\,5\,1\,4\,6$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$\sigma = 3\,2\,5\,1\,4\,6$

$2\,3\,5\,1\,4\,6$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow \mathit{length}(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $\mathit{swap}\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$2\,3\,5\,1\,4\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
  $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
  if $\sigma[j] > \sigma[j+1]$ then
    $swap(\sigma[j], \sigma[j+1])$
  end if
end for

Example

$\sigma = 3\,2\,5\,1\,4\,6$
$2\,3\,1\,5\,4\,6$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$2\,3\,1\,5\,4\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$\sigma = 3\,2\,5\,1\,4\,6$

$2\,3\,1\,4\,5\,6$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
  $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
  if $\sigma[j] > \sigma[j+1]$ then
    $swap\,(\sigma[j], \sigma[j+1])$
  end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$2\,3\,1\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

    $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

    if $\sigma[j] > \sigma[j+1]$ then

        $swap\,(\sigma[j], \sigma[j+1])$

    end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,3\,1\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,3\,1\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,3\,1\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,1\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,1\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,1\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$2\,1\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap\,(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$B^2(\sigma) = 2\,1\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\begin{aligned} \sigma &= 3\,2\,5\,1\,4\,6 \\ B(\sigma) &= 2\,3\,1\,4\,5\,6 \\ B^2(\sigma) &= 2\,1\,3\,4\,5\,6 \\ &\ 2\,1\,3\,4\,5\,6 \end{aligned}$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\begin{aligned} \sigma &= 3\,2\,5\,1\,4\,6 \\ B(\sigma) &= 2\,3\,1\,4\,5\,6 \\ B^2(\sigma) &= 2\,1\,3\,4\,5\,6 \\ & \quad 2\,1\,3\,4\,5\,6 \end{aligned}$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\begin{aligned} \sigma &= 3\,2\,5\,1\,4\,6 \\ B(\sigma) &= 2\,3\,1\,4\,5\,6 \\ B^2(\sigma) &= 2\,1\,3\,4\,5\,6 \\ & \ 1\,2\,3\,4\,5\,6 \end{aligned}$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$B^2(\sigma) = 2\,1\,3\,4\,5\,6$$
$$1\,2\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

  $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

  if $\sigma[j] > \sigma[j+1]$ then

    $swap(\sigma[j], \sigma[j+1])$

  end if

end for

Example

$$\begin{aligned} \sigma &= 3\,2\,5\,1\,4\,6 \\ B(\sigma) &= 2\,3\,1\,4\,5\,6 \\ B^2(\sigma) &= 2\,1\,3\,4\,5\,6 \\ & \ 1\,2\,3\,4\,5\,6 \end{aligned}$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap\,(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$B^2(\sigma) = 2\,1\,3\,4\,5\,6$$
$$1\,2\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \rightarrow \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    $\sigma \leftarrow B(\sigma)$
end for

Bubblesort Procedure

Procedure B : $\sigma \rightarrow \sigma$

for j from 1 to $n-1$
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

Example

$$\sigma = 3\,2\,5\,1\,4\,6$$
$$B(\sigma) = 2\,3\,1\,4\,5\,6$$
$$B^2(\sigma) = 2\,1\,3\,4\,5\,6$$
$$1\,2\,3\,4\,5\,6$$

Bubblesort

Bubblesort Algorithm

Bubblesort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$

for i from 1 to $n-1$

    $\sigma \leftarrow B(\sigma)$

end for

Bubblesort Procedure

Procedure B : $\sigma \to \sigma$

for j from 1 to $n-1$

    if $\sigma[j] > \sigma[j+1]$ then

        $swap\,(\sigma[j], \sigma[j+1])$

    end if

end for

Example

$$\begin{aligned} \sigma &= 3\,2\,5\,1\,4\,6 \\ B(\sigma) &= 2\,3\,1\,4\,5\,6 \\ B^2(\sigma) &= 2\,1\,3\,4\,5\,6 \\ B^3(\sigma) &= 1\,2\,3\,4\,5\,6 \end{aligned}$$

Dual Bubblesort

Cocktail Shaker Sort

Cocktail Shaker Sort Algorithm : $\sigma \to \sigma$

$n \leftarrow length(\sigma)$
for i from 1 to $n-1$
    if $i \bmod 2 = 0$ then
        $\sigma \leftarrow B(\sigma)$
    else
        $\sigma \leftarrow \tilde{B}(\sigma)$
    end if
end for

Dual Bubblesort

Procedure $\tilde{B}$: $\sigma \to \sigma$

for j from $n-1$ to 1
    if $\sigma[j] > \sigma[j+1]$ then
        $swap(\sigma[j], \sigma[j+1])$
    end if
end for

$$\tilde{B} = \rho \circ B \circ \rho$$

Dual sorting procedures

Dual procedure $\tilde{X}$

$$\tilde{X} = \rho \circ X \circ \rho$$

Proposition

$$\tilde{X}^{n-1}(\sigma) = id, \qquad \forall\, \sigma \in \Sigma_n$$

Hybrid algorithms

Procedures

$$\mathcal{P} = \{B, S, D^{ir}, D^{or}, \tilde{B}, \tilde{S}, \tilde{D^{ir}}, \tilde{D^{or}}\}$$

Hybrid sorting algorithm

If P_1, P_2, ... $P_{n-1} \in \mathcal{P}$, then

$$P_1 \circ P_2 \circ \cdots \circ P_{n-1}(\sigma) = id, \qquad \forall\, \sigma \in \Sigma_n$$

Problem

For which pair of procedures (P_1, P_2)

$$P_1 \circ P_2 = P_2 \circ P_1?$$

Commutativity

	B	$\tilde{B}$	S	$\tilde{S}$	D^{ir}	$\tilde{D}^{ir}$	D^{or}
$\tilde{B}$	Yes						
S	No (4231)	Yes					
$\tilde{S}$	Yes	No (4231)	No (4312)				
D^{ir}	No (34251)	? (Open)	No (43251)	No (53421)			
$\tilde{D}^{ir}$	? (Open)	No (51423)	No (54231)	No (51432)	No (5463271)		
D^{or}	No (53241)	? (Open)	No (53241)	No (53421)	No (634251)	No (645132)	
$\tilde{D}^{or}$	? (Open)	No (52431)	No (54231)	No (52431)	No (546231)	No (625341)	No (465231)

A bijective enumeration of $Sort_n(\mathbb{X})$

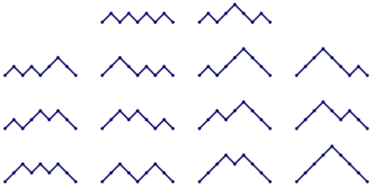
A bijective enumeration of $Sort_n(\mathbb{X})$

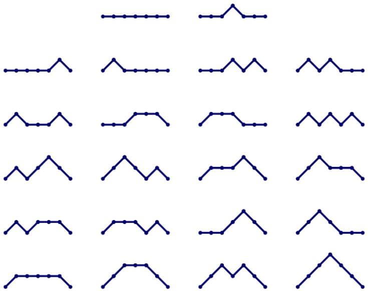
Catalan numbers

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

Schröder numbers

$$\begin{cases} S_n = S_{n-1} + \sum_{i=0}^{n-1} S_i \, S_{n-1-i} \\ S_0 = 1 \end{cases}$$





A bijective enumeration of $Sort_n(\mathbb{X})$

Open problem

$$|Sort_n(\mathbb{D})| = ?$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Open problem

$$|Sort_n(\mathbb{D})| = ?$$

Enumerative results

$$\begin{aligned} |Sort_n(\mathbb{S})| &= C_n \\ |Sort_n(\mathbb{D}^{ir})| &= S_{n-1} \\ |Sort_n(\mathbb{D}^{or})| &= S_{n-1} \end{aligned}$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Open problem

$$|Sort_n(\mathbb{D})| = ?$$

Enumerative results

$$\begin{aligned} |Sort_n(\mathbb{S})| &= C_n &= |\mathscr{D}_{2n}| \\ |Sort_n(\mathbb{D}^{ir})| &= S_{n-1} &= |\mathscr{S}_{2(n-1)}| \\ |Sort_n(\mathbb{D}^{or})| &= S_{n-1} &= |\mathscr{S}_{2(n-1)}| \end{aligned}$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Open problem

$$|Sort_n(\mathbb{D})| = ?$$

Enumerative results

$$\begin{aligned}
|Sort_n(\mathbb{S})| &= C_n &= |\mathscr{D}_{2n}| \\
|Sort_n(\mathbb{D}^{ir})| &= S_{n-1} &= |\mathscr{S}_{2(n-1)}| \\
|Sort_n(\mathbb{D}^{or})| &= S_{n-1} &= |\mathscr{S}_{2(n-1)}|
\end{aligned}$$

Bijections

$$\begin{aligned}
Sort_n(\mathbb{S}) &\longleftrightarrow \mathscr{D}_{2n} \\
Sort_n(\mathbb{D}^{ir}) &\longleftrightarrow \mathscr{S}_{2(n-1)} \\
Sort_n(\mathbb{D}^{or}) &\longleftrightarrow \mathscr{S}_{2(n-1)}
\end{aligned}$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Bijections

$$
\begin{aligned}
Sort_n(\mathbb{S}) &\longleftrightarrow \mathscr{D}_{2n} \\
Sort_n(\mathbb{D}^{ir}) &\longleftrightarrow \mathscr{S}_{2(n-1)} \\
Sort_n(\mathbb{D}^{or}) &\longleftrightarrow \mathscr{S}_{2(n-1)}
\end{aligned}
$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Bijections

$$
\begin{aligned}
Sort_n(\mathbb{S}) &\longleftrightarrow \mathscr{D}_{2n}\\
Sort_n(\mathbb{D}^{ir}) &\longleftrightarrow \mathscr{S}_{2(n-1)}\\
Sort_n(\mathbb{D}^{or}) &\longleftrightarrow \mathscr{S}_{2(n-1)}
\end{aligned}
$$

Bijections

$$Sort(\mathbb{X}) \longleftrightarrow \mathscr{L}$$

A bijective enumeration of $Sort_n(\mathbb{X})$

Bijections

$$\begin{aligned} Sort_n(\mathbb{S}) &\longleftrightarrow \mathscr{D}_{2n} \\ Sort_n(\mathbb{D}^{ir}) &\longleftrightarrow \mathscr{S}_{2(n-1)} \\ Sort_n(\mathbb{D}^{or}) &\longleftrightarrow \mathscr{S}_{2(n-1)} \end{aligned}$$

Bijections

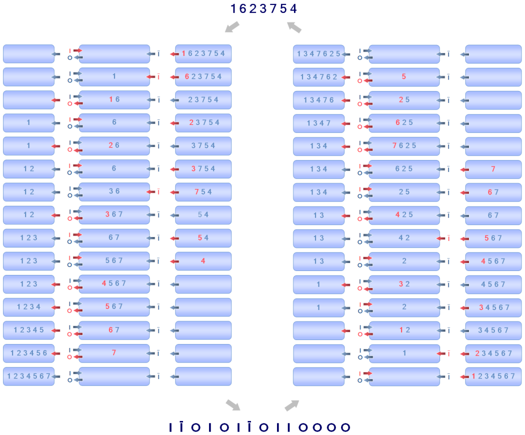
$$Sort(\mathbb{X}) \longleftrightarrow \mathscr{L}$$

Bijections

$$Sort(\mathbb{X}) \underset{\varphi_{\mathbb{X}}^{-1}}{\overset{\varphi_{\mathbb{X}}}{\rightleftarrows}} \bar{\mathcal{X}} \underset{\psi_{\mathbb{X}}^{-1}}{\overset{\psi_{\mathbb{X}}}{\rightleftarrows}} \mathscr{L}$$

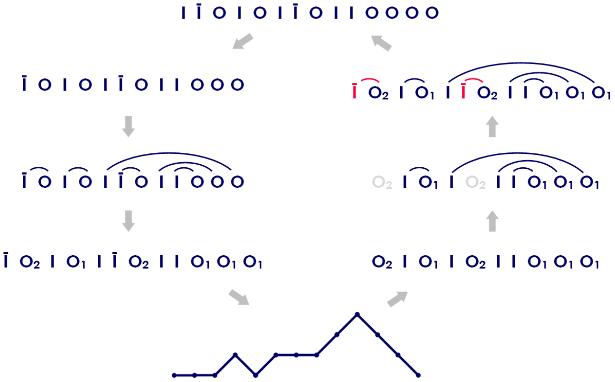
The bijection $\varphi_{\mathbb{X}}$ $(\mathbb{X} = \mathbb{D}^{or})$

$$Sort(\mathbb{X}) \underset{\varphi_{\mathbb{X}}^{-1}}{\overset{\varphi_{\mathbb{X}}}{\rightleftarrows}} \bar{\mathcal{X}}$$

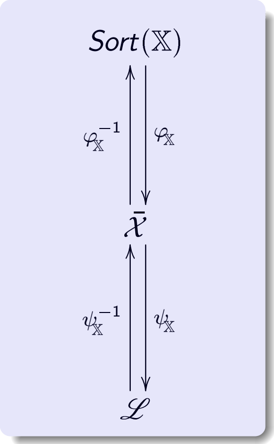


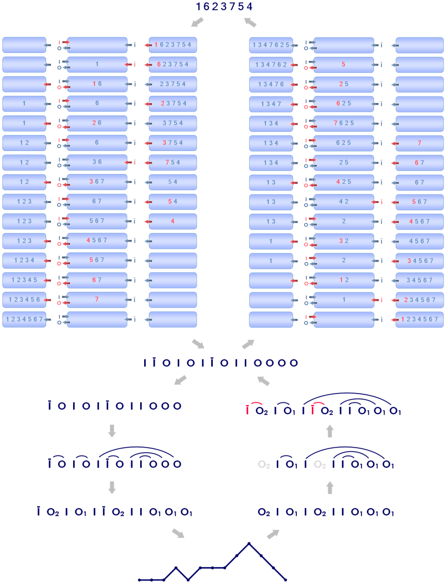
The bijection $\psi_{\mathbb{X}}$ ($\mathbb{X} = \mathbb{D}^{or}$)

$$\bar{\mathcal{X}} \underset{\psi_{\mathbb{X}}^{-1}}{\overset{\psi_{\mathbb{X}}}{\rightleftarrows}} \mathscr{L}$$



The bijection $\psi_{\mathbb{X}} \circ \varphi_{\mathbb{X}}$ $(\mathbb{X} = \mathbb{D}^{or})$





Deque sortable permutations and deterministic sorting procedures

Luca S. Ferrari

Dipartimento di Matematica
Università di Bologna

Permutation Patterns 2013
Paris