

Commutation properties among bubblesort, stacksort and their duals

Luca S. Ferrari



Dipartimento di Matematica
Università di Bologna

GASCom 2012
Bordeaux

Focus

Sorting algorithms \leftrightarrow Pattern avoidance

Background

Focus

Sorting algorithms \leftrightarrow Pattern avoidance

Theorem (Knuth, 1973)

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

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$$\sigma = 325146$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
325146

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
325146

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
235146

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
235146

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
235146

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
231546

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
231546

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
231456

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$\sigma = 325146$
231456

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\sigma = 325146$$
$$B(\sigma) = 231456$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ &231456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ &\quad \color{red}231456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ & \quad 2\mathbf{3}1456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ &\quad 213456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ & \quad 213456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ &213456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ &2134\mathbf{56}\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &213456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &\quad \mathbf{21}3456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &\quad \mathbf{1}23456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ & \quad 123456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &\quad 123456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &123456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ &123456\end{aligned}$$

Bubblesort

The algorithm

```
repeat
  for  $i$  from 1 to  $|\sigma| - 1$ 
    if  $\sigma_i > \sigma_{i+1}$ 
      swap( $\sigma_i, \sigma_{i+1}$ )
    end
  end
end
until  $\sigma$  is sorted
```

} B

Example

$$\begin{aligned}\sigma &= 325146 \\ B(\sigma) &= 231456 \\ B^2(\sigma) &= 213456 \\ B^3(\sigma) &= 123456\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\sigma = 523849176$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\sigma = 523849176$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= \end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 5 \quad 849176 \\ B(\sigma) &= 23\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= \quad 849176 \\ B(\sigma) &= 235\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= \quad 8 \quad 9 \quad 1 \quad 7 \quad 6 \\ B(\sigma) &= 2 \quad 3 \quad 5 \quad 4\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= \quad \quad \quad 9176 \\ B(\sigma) &= 23548\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned} \sigma &= \quad \quad \quad 9 \\ B(\sigma) &= 23548176 \end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned} \sigma &= \\ B(\sigma) &= 23\mathbf{5}4\mathbf{8}176\mathbf{9} \end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\sigma = 523849176$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\sigma = 523849176$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{array}{ccccccccc}\sigma &= & 5 & 2 & 3 & 8 & 4 & & 1 & 7 & 6 \\ & & & & & & & & & & 9\end{array}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 52384 \\ &\quad 1769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 52384 \\ &\quad 1769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 523 \quad 4 \\ &\quad \quad \quad 81769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 523 \\ &481769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 523 \\ &481769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 23 \\ &5481769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= \\ &235481769\end{aligned}$$

Bubblesort

Left-to-right maxima definition

$$B(M_1 u_1 M_2 u_2 \cdots M_k u_k) = u_1 M_1 u_2 M_2 \cdots u_k M_k$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Recursive definition

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Example

$$\begin{aligned}\sigma &= 523849176 \\ B(\sigma) &= 235481769\end{aligned}$$

Stacksort


The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until  $output$  is sorted
```


Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until  $output$  is sorted
```

} S

Example

$S(\sigma)$ output

input σ

stack

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

534261



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

5 3 4 2 6 1



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example



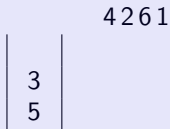
Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3 4 2 6 1

 |

 5

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3

261

4

5

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3

2 6 1

4

5

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3

2
4
5

61

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until  $output$  is sorted
```

} S

Example

3

2
4
5

6 1

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

32

61

4

5

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3 2 4

6 1

5

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3 2 4 5


6 1



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```



Example

3 2 4 5 1



6

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

}

S

Example

3 2 4 5

1

6

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3 2 4 5

1
6

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until output is sorted
```

} S

Example

3 2 4 5 1

6

Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until  $output$  is sorted
```

} S

Example

3 2 4 5 1 6



Stacksort

The algorithm

```
repeat
  repeat
    if  $first(input) < top(stack) \vee stack == \emptyset$ 
      push
    else
      pop
    end
  until  $input \cup stack == \emptyset$ 
until  $output$  is sorted
```

} S

Example

$S(\sigma) = 324516$

$\sigma = 534261$



Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\sigma = 537921486$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\sigma = 537921486$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\sigma = 537921486$$
$$S(\sigma) =$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 37921486 \\ S(\sigma) &= 5\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= \quad 7921486 \\ S(\sigma) &= 35\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= \quad 921486 \\ S(\sigma) &= 357\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{array}{rcl} \sigma = & & 21486 \\ S(\sigma) = & 357 & 9 \end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned} \sigma &= && 2 & 1 & 4 & 8 & 6 \\ S(\sigma) &= & 3 & 5 & 7 & & & 9 \end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned} \sigma &= && 1 & 4 & 8 & 6 \\ S(\sigma) &= & 3 & 5 & 7 & 2 & 9 \end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{array}{rcl} \sigma = & & 486 \\ S(\sigma) = & 35712 & 9 \end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{array}{r} \sigma = \qquad \qquad \qquad 86 \\ S(\sigma) = 357124 \quad 9 \end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{array}{r} \sigma = \qquad \qquad \qquad 6 \\ S(\sigma) = 3 \mathbf{57124} \mathbf{89} \end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\sigma =$$
$$S(\sigma) = 3\mathbf{5}71\mathbf{2}46\mathbf{8}9$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\sigma = 537921486$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\sigma = 537\mathbf{9}21486$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= 537 \quad 21486 \\ &\qquad\qquad\qquad 9\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= 537 \quad 21486 \\ &\qquad\qquad\qquad 9\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{array}{ccccc}\sigma &= & 53 & & 21486 \\ & & & 7 & & 9\end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{array}{rcc}\sigma = 53 & 214 & 6 \\ & 7 & 89\end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{array}{ccccc}\sigma = & 53 & & 214 & 6 \\ & & & 7 & 89\end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{array}{cccc}\sigma = & 3 & 214 & 6 \\ & 57 & & 89\end{array}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= \begin{array}{ccc} 3 & 21 & 6 \\ 57 & 4 & 89 \end{array}\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= & 3 & 21 \\ & 57 & 4689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= \quad 3 \quad 21 \\ &\quad 57 \quad 4689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= \quad \quad 21 \\ &\quad 357 \quad 4689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= && 1 \\ &357 & 24689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= \quad \quad \quad 1 \\ & \quad \quad \quad 357 \quad 24689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= \\ &357124689\end{aligned}$$

Left-to-right maxima definition

$$S(M_1 u_1 \cdots M_k u_k) = S(u_1) M_1 \cdots S(u_k) M_k$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Recursive definition

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Example

$$\begin{aligned}\sigma &= 537921486 \\ S(\sigma) &= 357124689\end{aligned}$$

Bubblesort and stacksort

Proposition

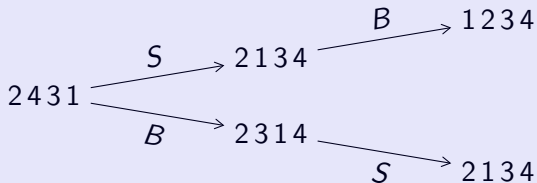
$$B \circ S \neq S \circ B$$

Bubblesort and stacksort

Proposition

$$B \circ S \neq S \circ B$$

Proof



Dual sorting algorithms

Dual sorting algorithms

Reverse-complement operator

$$\rho = r \circ c$$

Dual sorting algorithms

Reverse-complement operator

$$\rho = r \circ c$$

Example

$$\begin{array}{ccc} \sigma = 32641875 & & \\ \downarrow c(\sigma)_i = n+1-\sigma_i & & \\ c(\sigma) = 67358124 & & \\ \downarrow r[c(\sigma)]_i = c(\sigma)_{n+1-i} & & \\ r \circ c(\sigma) = 42185376 & & \end{array}$$

Dual sorting algorithms

Dual bubblesort

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

Dual sorting algorithms

Dual bubblesort

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

Dual stacksort

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

Dual sorting algorithms

Dual bubblesort

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

Dual stacksort

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

Remark

Left-to-right maxima $\xleftrightarrow{\rho}$ Right-to-left minima

Overview

Bubblesort

σ read from left to right

swap $\iff \sigma_i > \sigma_{i+1}$

$$(M_i u_i)_i \xrightarrow{B} (u_i M_i)_i$$

$$B(\alpha n \beta) = B(\alpha) \beta n$$

Dual bubblesort

σ read from right to left

swap $\iff \sigma_i > \sigma_{i+1}$

$$(v_j m_j)_j \xrightarrow{\tilde{B}} (m_j v_j)_j$$

$$\tilde{B}(\tau 1 \omega) = 1 \tau \tilde{B}(\omega)$$

Stacksort

output

input



pop \iff input $>$ stack

$$(M_i u_i)_i \xrightarrow{S} (S(u_i) M_i)_i$$

$$S(\alpha n \beta) = S(\alpha) S(\beta) n$$

Dual stacksort

input

output



pop \iff input $<$ stack

$$(v_j m_j)_j \xrightarrow{\tilde{S}} (m_j \tilde{S}(v_j))_j$$

$$\tilde{S}(\tau 1 \omega) = 1 \tilde{S}(\tau) \tilde{S}(\omega)$$

Commutation results

Commutation results

Theorem

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

Commutation results

Theorem

- $B \circ \tilde{B} = \tilde{B} \circ B$
- $S \circ \tilde{B} = \tilde{B} \circ S$

Commutation results

Theorem

- $B \circ \tilde{B} = \tilde{B} \circ B$
- $S \circ \tilde{B} = \tilde{B} \circ S$
- $\tilde{S} \circ B = B \circ \tilde{S}$

Commutation results

Theorem

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

- $S \circ \tilde{B} = \tilde{B} \circ S$

- $\tilde{S} \circ B = B \circ \tilde{S}$

$$(B \circ \rho)^2 = (\rho \circ B)^2$$

$$S \circ \rho \circ B \circ \rho = \rho \circ B \circ \rho \circ S$$

$$\rho \circ S \circ \rho \circ B = B \circ \rho \circ S \circ \rho$$

Commutation results

Theorem

- $B \circ \tilde{B} = \tilde{B} \circ B$ $(B \circ \rho)^2 = (\rho \circ B)^2$
- $S \circ \tilde{B} = \tilde{B} \circ S$ $S \circ \rho \circ B \circ \rho = \rho \circ B \circ \rho \circ S$
- $\tilde{S} \circ B = B \circ \tilde{S}$ $\rho \circ S \circ \rho \circ B = B \circ \rho \circ S \circ \rho$

Sketch of the proof

Introduce the *local sorting operators*

$$\begin{array}{ccc} M_1 u_1 \cdots M_i u_i \cdots M_k u_k & \xrightarrow{B_{M_i}} & M_1 u_1 \cdots u_i M_i \cdots M_k u_k \\ v_h m_h \cdots v_j m_j \cdots v_1 m_1 & \xrightarrow{\tilde{B}_{m_j}} & v_h m_h \cdots m_j v_j \cdots v_1 m_1 \end{array}$$

Commutation results

Theorem

- $B \circ \tilde{B} = \tilde{B} \circ B$ $(B \circ \rho)^2 = (\rho \circ B)^2$
- $S \circ \tilde{B} = \tilde{B} \circ S$ $S \circ \rho \circ B \circ \rho = \rho \circ B \circ \rho \circ S$
- $\tilde{S} \circ B = B \circ \tilde{S}$ $\rho \circ S \circ \rho \circ B = B \circ \rho \circ S \circ \rho$

Sketch of the proof

Introduce the *local sorting operators*

$$\begin{array}{ccc} M_1 u_1 \cdots M_i u_i \cdots M_k u_k & \xrightarrow{B_{M_i}} & M_1 u_1 \cdots u_i M_i \cdots M_k u_k \\ v_h m_h \cdots v_j m_j \cdots v_1 m_1 & \xrightarrow{\tilde{B}_{m_j}} & v_h m_h \cdots m_j v_j \cdots v_1 m_1 \end{array}$$

$$B = B_{M_1} \circ B_{M_2} \circ \cdots \circ B_{M_k}$$

$$\tilde{B} = \tilde{B}_{m_1} \circ \tilde{B}_{m_2} \circ \cdots \circ \tilde{B}_{m_h}$$

Commutation results

Theorem

- $B \circ \tilde{B} = \tilde{B} \circ B$ $(B \circ \rho)^2 = (\rho \circ B)^2$
- $S \circ \tilde{B} = \tilde{B} \circ S$ $S \circ \rho \circ B \circ \rho = \rho \circ B \circ \rho \circ S$
- $\tilde{S} \circ B = B \circ \tilde{S}$ $\rho \circ S \circ \rho \circ B = B \circ \rho \circ S \circ \rho$

Sketch of the proof

Introduce the *local sorting operators*

$$\begin{array}{ccc} M_1 u_1 \cdots M_i u_i \cdots M_k u_k & \xrightarrow{B_{M_i}} & M_1 u_1 \cdots u_i M_i \cdots M_k u_k \\ v_h m_h \cdots v_j m_j \cdots v_1 m_1 & \xrightarrow{\tilde{B}_{m_j}} & v_h m_h \cdots m_j v_j \cdots v_1 m_1 \end{array}$$

$$B = B_{M_1} \circ B_{M_2} \circ \cdots \circ B_{M_k}$$

$$\tilde{B} = \tilde{B}_{m_1} \circ \tilde{B}_{m_2} \circ \cdots \circ \tilde{B}_{m_h}$$

$$B \circ \tilde{B} = \tilde{B} \circ B \iff B_{M_i} \circ \tilde{B}_{m_j} = \tilde{B}_{m_j} \circ B_{M_i} \quad \forall i, j$$

$$S \circ \tilde{B} = \tilde{B} \circ S \iff S \circ \tilde{B}_{m_j} = \tilde{B}_{m_j} \circ S \quad \forall j$$

Commutation results

Proposition

- $B \circ S \neq S \circ B$

Commutation results

Proposition

- $B \circ S \neq S \circ B$
- $\tilde{B} \circ \tilde{S} \neq \tilde{S} \circ \tilde{B}$

Commutation results

Proposition

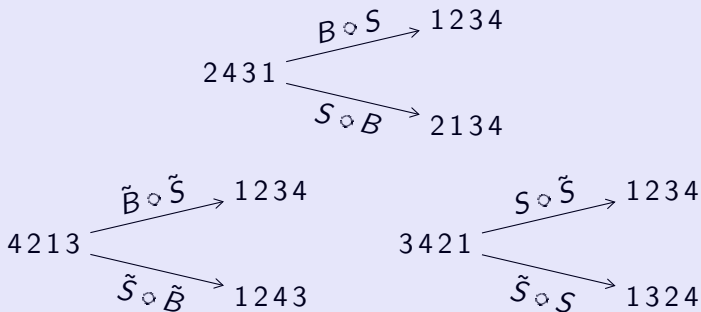
- $B \circ S \neq S \circ B$
- $\tilde{B} \circ \tilde{S} \neq \tilde{S} \circ \tilde{B}$
- $S \circ \tilde{S} \neq \tilde{S} \circ S$

Commutation results

Proposition

- $B \circ S \neq S \circ B$
- $\tilde{B} \circ \tilde{S} \neq \tilde{S} \circ \tilde{B}$
- $S \circ \tilde{S} \neq \tilde{S} \circ S$

Proof



Pattern avoidance

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Avoiding a set of pattern

$$Av(\tau_1 \cdots \tau_k) = \bigcap_{i=1}^k Av(\tau_i)$$

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Avoiding a set of pattern

$$Av(\tau_1 \cdots \tau_k) = \bigcap_{i=1}^k Av(\tau_i)$$

Example

$$\sigma = 215436$$

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Avoiding a set of pattern

$$Av(\tau_1 \cdots \tau_k) = \bigcap_{i=1}^k Av(\tau_i)$$

Example

$$\sigma = 215436$$

$$\sigma \in Av(231, 312)$$

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Avoiding a set of pattern

$$Av(\tau_1 \cdots \tau_k) = \bigcap_{i=1}^k Av(\tau_i)$$

Example

$$\sigma = 215436$$

$$\sigma \in Av(231, 312)$$

$$\sigma \notin Av(132)$$

Pattern avoidance

Definition

σ avoids a pattern τ
($\sigma \in Av(\tau)$) \iff σ has no subsequences
order-isomorphic to τ

Avoiding a set of pattern

$$Av(\tau_1 \cdots \tau_k) = \bigcap_{i=1}^k Av(\tau_i)$$

Example

$\sigma = 215436$

$\sigma \in Av(231, 312)$

$\sigma \notin Av(132)$

Sorting algorithms and pattern avoidance

Permutations sorted by A

$$\text{Sort}(A) = \{\sigma \mid A(\sigma) = 12 \cdots n\}$$

Sorting algorithms and pattern avoidance

Permutations sorted by A

$$\text{Sort}(A) = \{\sigma \mid A(\sigma) = 12 \cdots n\}$$

Theorem (Knuth, 1973)

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

Sorting algorithms and pattern avoidance

Permutations sorted by A

$$\text{Sort}(A) = \{\sigma \mid A(\sigma) = 12 \cdots n\}$$

Theorem (Knuth, 1973)

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

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B) = \text{Av}(231, 321)$$

$$\text{Sort}(S \circ B) = \text{Av}(2341, 2431, 3241, 4231)$$

Sorting algorithms and pattern avoidance

Permutations sorted by A

$$\text{Sort}(A) = \{\sigma \mid A(\sigma) = 12 \cdots n\}$$

Theorem (Knuth, 1973)

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

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B) = \text{Av}(231, 321)$$

$$\text{Sort}(S \circ B) = \text{Av}(2341, 2431, 3241, 4231)$$

Theorem (Barnabei, Bonetti and Silimbani, 2012)

$$\text{Sort}(\tilde{B} \circ B) = \text{Av}(3412, 3421, 4312, 4321)$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] =$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$3\ 1\ 2 [21, 213, 132] =$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 213$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 213$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 213465$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 213465$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to α_i , $\forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j$, $\forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[\ell_1, \dots, \ell_n] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[[\ell_1, \dots, \ell_n]] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Example

$$231[[2, 1, 2]] = \{34512, 34521, 43512, 43521\}$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to α_i , $\forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j$, $\forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[\ell_1, \dots, \ell_n] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Example

$$231[[2, 1, 2]] = \{34512, 34521, 43512, 43521\}$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[\ell_1, \dots, \ell_n] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Example

$$231[[2, 1, 2]] = \{34512, 34521, 43512, 43521\}$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[\ell_1, \dots, \ell_n] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Example

$$231[[2, 1, 2]] = \{34512, 34521, 43512, 43521\}$$

Sorting algorithms and pattern avoidance

Inflation of σ by $\alpha_1, \alpha_2, \dots, \alpha_n$

$$\sigma = \sigma_1 \sigma_2 \dots \sigma_n$$

$$\sigma[\alpha_1, \alpha_2, \dots, \alpha_n] = p_1 p_2 \dots p_n$$

- p_i order-isomorphic to $\alpha_i, \quad \forall i$
- $p_i < p_j \iff \sigma_i < \sigma_j, \quad \forall i, j$

Example

$$312[21, 213, 132] = 87213465$$

Inflations of σ by α_i of fixed length

$$\sigma[[\ell_1, \dots, \ell_n]] = \{\sigma[\alpha_1, \dots, \alpha_n] : |\alpha_i| = \ell_i, \forall i\}$$

Example

$$231[[2, 1, 2]] = \{34512, 34521, 43512, 43521\}$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Theorem

$$\text{Sort}(B^h \circ \tilde{B}^k) = \text{Av}(21 \llbracket h + 1, k + 1 \rrbracket)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Theorem

$$\text{Sort}(B^h \circ \tilde{B}^k) = \text{Av}(21 \llbracket h + 1, k + 1 \rrbracket)$$

Example

$$\text{Sort}(B \circ \tilde{B}) = \text{Av}(21 \llbracket 2, 2 \rrbracket) = \text{Av}(3412, 3421, 4312, 4321)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Theorem

$$\text{Sort}(B^h \circ \tilde{B}^k) = \text{Av}(21 \llbracket h + 1, k + 1 \rrbracket)$$

Example

$$\text{Sort}(B \circ \tilde{B}) = \text{Av}(21 \llbracket 2, 2 \rrbracket) = \text{Av}(34\mathbf{1}2, 34\mathbf{2}1, 43\mathbf{1}2, 43\mathbf{2}1)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Theorem

$$\text{Sort}(B^h \circ \tilde{B}^k) = \text{Av}(21 \llbracket h + 1, k + 1 \rrbracket)$$

Example

$$\text{Sort}(B \circ \tilde{B}) = \text{Av}(21 \llbracket 2, 2 \rrbracket) = \text{Av}(3412, 3421, 4312, 4321)$$

Sorting algorithms and pattern avoidance

Theorem (Albert, Atkinson, Bouvel, Claesson and Dukes, 2011)

$$\text{Sort}(B^h) = \text{Av}(21 \llbracket h + 1, 1 \rrbracket)$$

Corollary

$$\text{Sort}(\tilde{B}^k) = \text{Av}(21 \llbracket 1, k + 1 \rrbracket)$$

Theorem

$$\text{Sort}(B^h \circ \tilde{B}^k) = \text{Av}(21 \llbracket h + 1, k + 1 \rrbracket)$$

Example

$$\text{Sort}(B \circ \tilde{B}) = \text{Av}(21 \llbracket 2, 2 \rrbracket) = \text{Av}(3412, 3421, 4312, 4321)$$

Commutation properties among bubblesort, stacksort and their duals

Luca S. Ferrari



Dipartimento di Matematica
Università di Bologna

GASCom 2012
Bordeaux