

# 1 Datenstrukturen

- 1.1 Abstrakte Datentypen
- 1.2 Lineare Strukturen
- 1.3 Bäume
- 1.4 **Prioritätsschlangen**
- 1.5 Graphen



# Prioritätsschlangen

- Warteschlangen mit „Vordrängeln“
- Einfügen am Ende der Schlange
- Jedes Element hat eine Priorität
- Deq() liefert immer das Element mit der höchsten Priorität



# Prioritätsschlangen

- Elemente:  $X \in W' = W \times R$   
Wertebereich:  $L = \{ \} \cup W' \cup W'^2 \cup W'^3 \cup \dots$
- Create :  $\rightarrow L$
- Enq\* :  $W' \times L \rightarrow L$
- Deq :  $L \rightarrow L$
- Get :  $L \rightarrow W'$
- Empty :  $L \rightarrow \text{Bool}$



# Prioritätsschlangen

- $\text{Empty}(\text{Create}()) = \text{true}$
- $\text{Empty}(\text{Enq}(x,z)) = \text{false}$
- $\text{Deq}(\text{Enq}(x,\text{Create}())) = \text{Create}()$
- $\text{Deq}(\text{Enq}(x,z)) = \text{Enq}(x,\text{Deq}(z)) \quad \text{if } z \neq \{ \}$
- $\text{Get}(\text{Enq}(x,\text{Create}())) = x$
- $\text{Get}(\text{Enq}(x,z)) = \text{Get}(z) \quad \text{if } z \neq \{ \}$



# Prioritätsschlangen

- $\text{Enq}()$  : Standardoperator
- $\text{Enq}^*()$  :  
Füge Element  $x$  bzgl. seiner Priorität  $\text{prio}[x]$  ein:

$$\begin{aligned} - \text{Enq}^*(x, \text{Enq}(y, z)) &= \text{Enq}(y, \text{Enq}^*(x, z)) \\ &\dots \text{ if } \text{prio}[x] > \text{prio}[y] \end{aligned}$$

$$\begin{aligned} - \text{Enq}^*(x, \text{Enq}(y, z)) &= \text{Enq}(x, \text{Enq}(y, z)) \\ &\dots \text{ if } \text{prio}[x] \leq \text{prio}[y] \end{aligned}$$



# Prioritätsschlangen

- Aufwand steigt proportional zur Länge der Liste (= Zahl der Axiom-Anwendungen).
- Liste wird vollständig sortiert, obwohl eigentlich nur das Element mit maximaler Priorität gefunden werden muss.
- Ein- und Ausfüge-Reihenfolge nicht vorhersagbar.



# Effiziente Implementierung

- Warteschlangen: einfache Implementierung (front und back Pointer)
- Binär-Bäume: schneller Elementzugriff
- Array-Implementierung vollständiger Bäume kombiniert beide Vorteile
- Aber: Bei Suchbäumen ist das maximale Element unten im Baum ...



# Heap-Bedingung

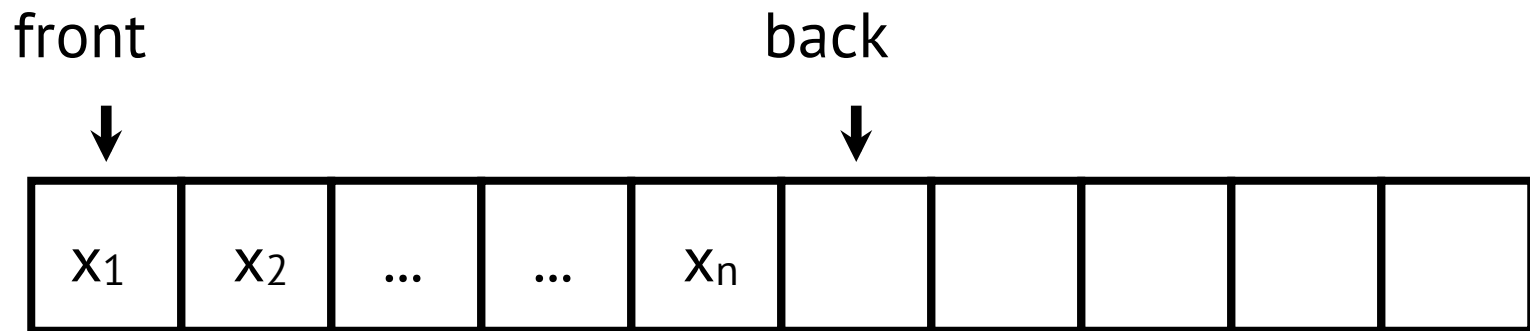
- Neue Sortierung der Knoten im Baum. Für alle Knoten  $T = \text{Node}(L,x,R)$  muss gelten
  - $\text{max\_prio}(L) \leq \text{prio}[x]$
  - $\text{max\_prio}(R) \leq \text{prio}[x]$
- Element mit maximaler Priorität ist im Wurzelknoten gespeichert.
- Sortierung der Elemente nur entlang der Pfade im Binärbaum.





# Effiziente Implementierung

- Array-Implementierung
  - Front-Pointer bleibt fest
  - Element mit maximaler Priorität in  $S[1]$
  - Back-Pointer variabel
  - Einfügen in  $S[\text{back}]$



# Effiziente Implementierung

- Enq(x)

```
if back = length+1 then  
  Q-FULL-ERROR
```

```
else
```

```
  i ← back  
  back ← back+1  
  S[ i ] ← x
```

```
  while i ≠ 1 and S[ i ] > S[ i / 2 ] do
```

```
    swap(S[ i ], S[ i / 2 ])
```

```
    i ← i / 2
```

# Effiziente Implementierung

- Deq()

```
if back = 1 then  
  Q-EMPTY-ERROR
```

```
else
```

```
  i ← 1; back ← back-1; S[1] ← S[back]
```

```
  while i ≤ (back-1) / 2 do
```

```
    j ← 2×i
```

```
    if j < back-1 and S[ j ] < S[ j+1 ] then
```

```
      j ← j+1
```

```
    if S[ i ] < S[ j ] then
```

```
      swap(S[ i ], S[ j ])
```

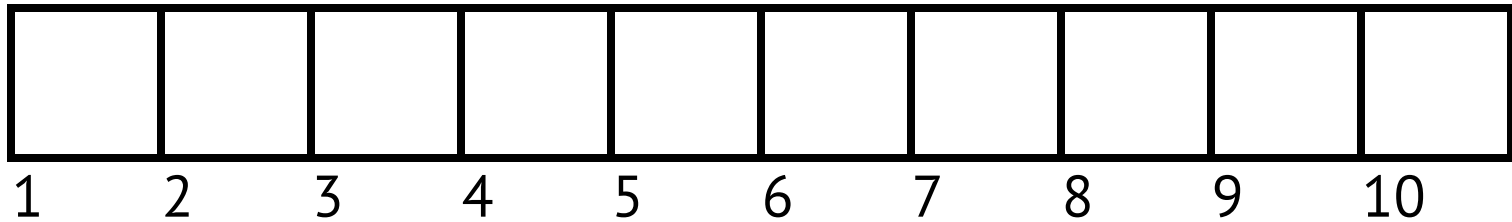
```
      i ← j
```

```
    else
```

```
      i ← back
```

# Beispiel

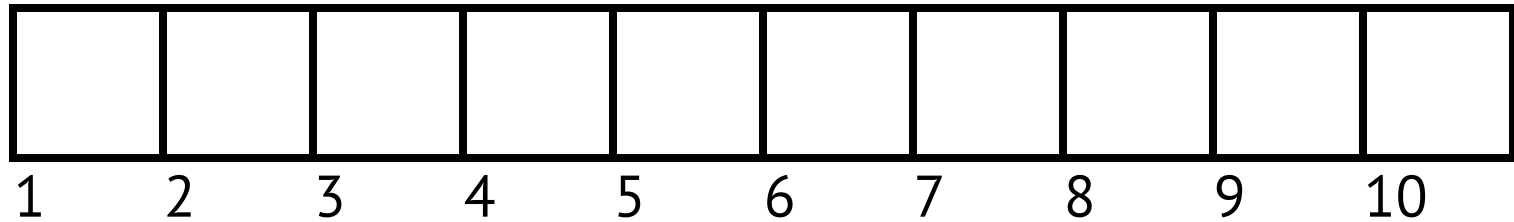
back



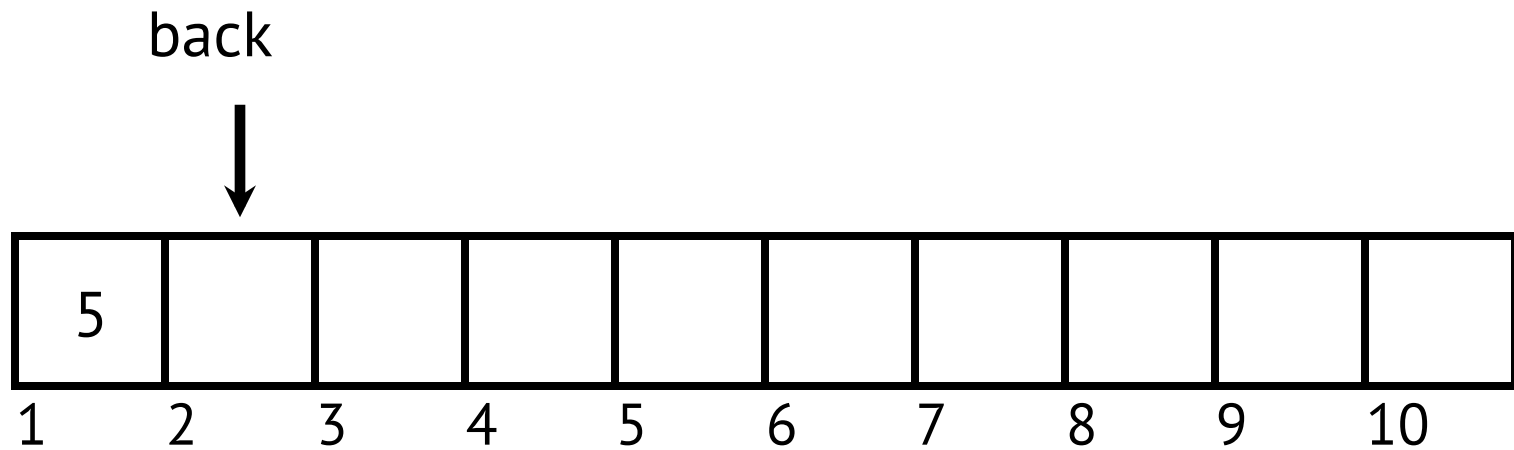
# Beispiel

Enq(5)

back



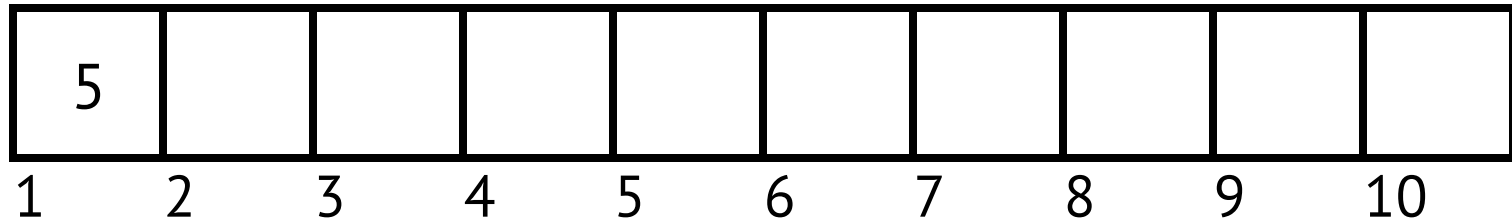
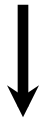
# Beispiel



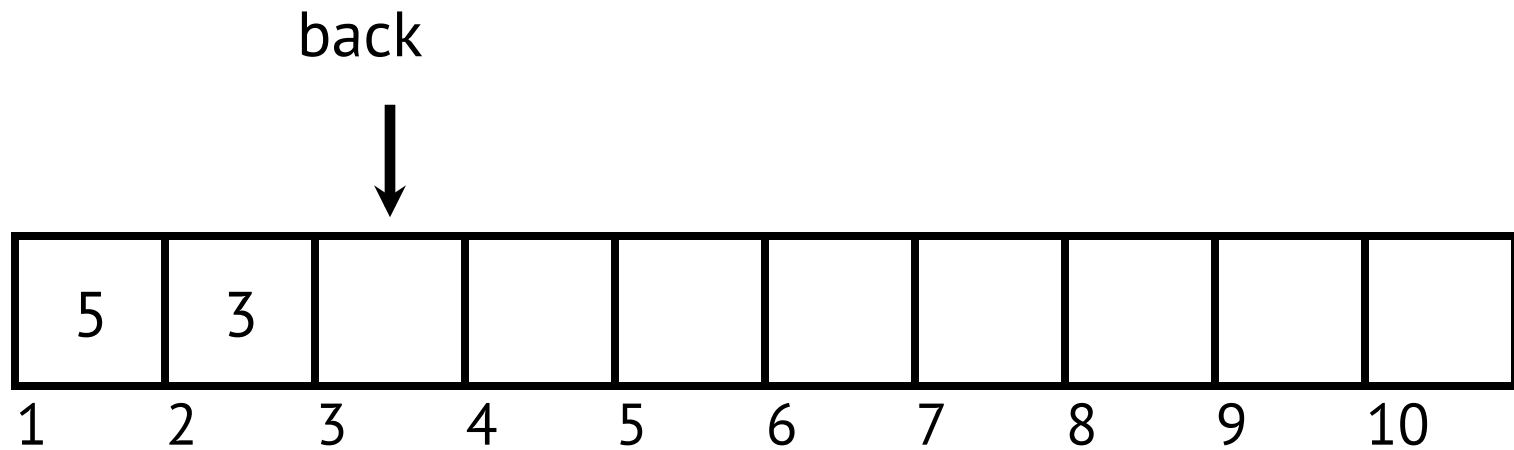
# Beispiel

Enq(3)

back



# Beispiel

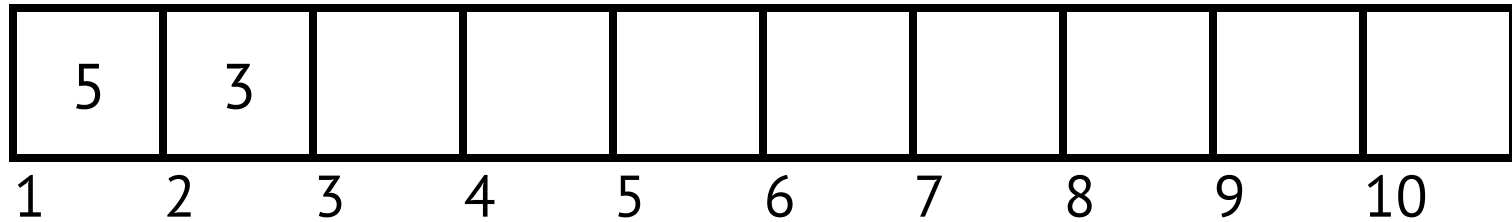




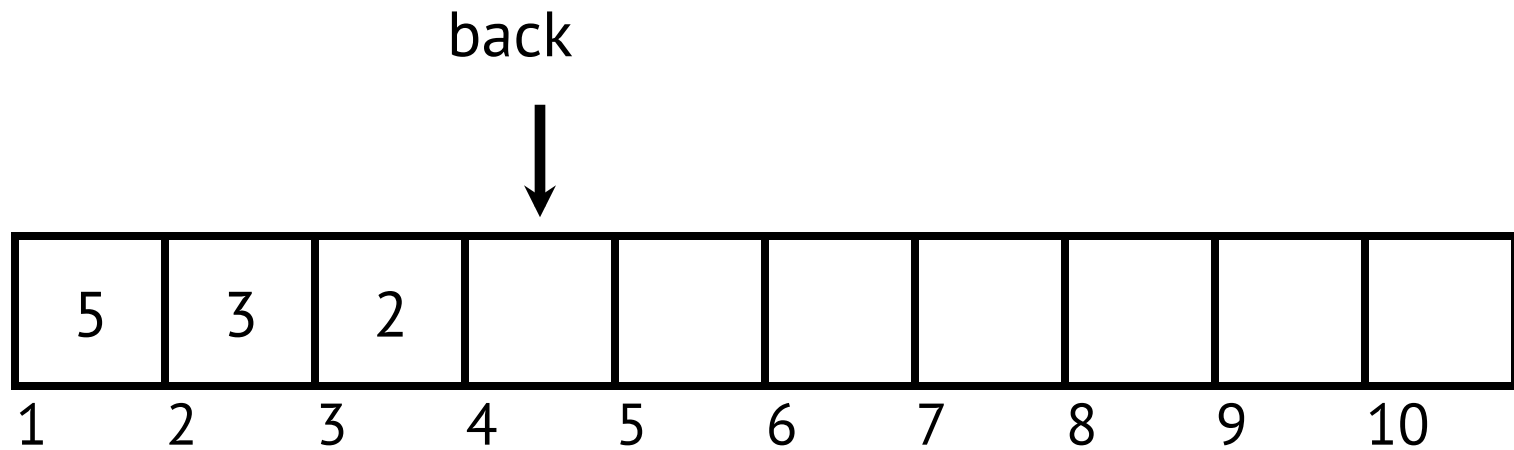
# Beispiel

Enq(2)

back



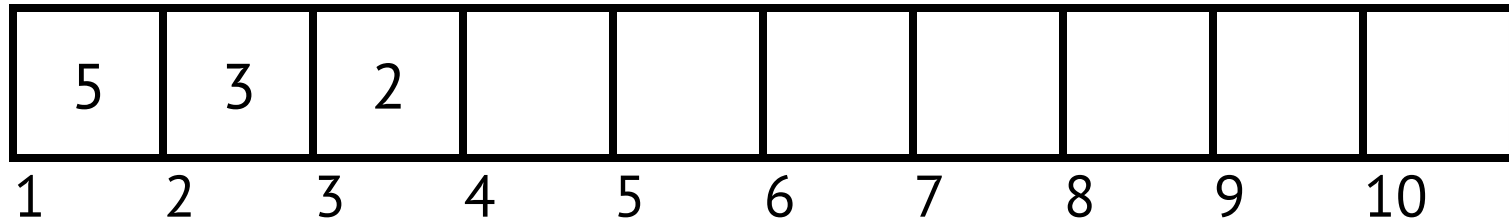
# Beispiel



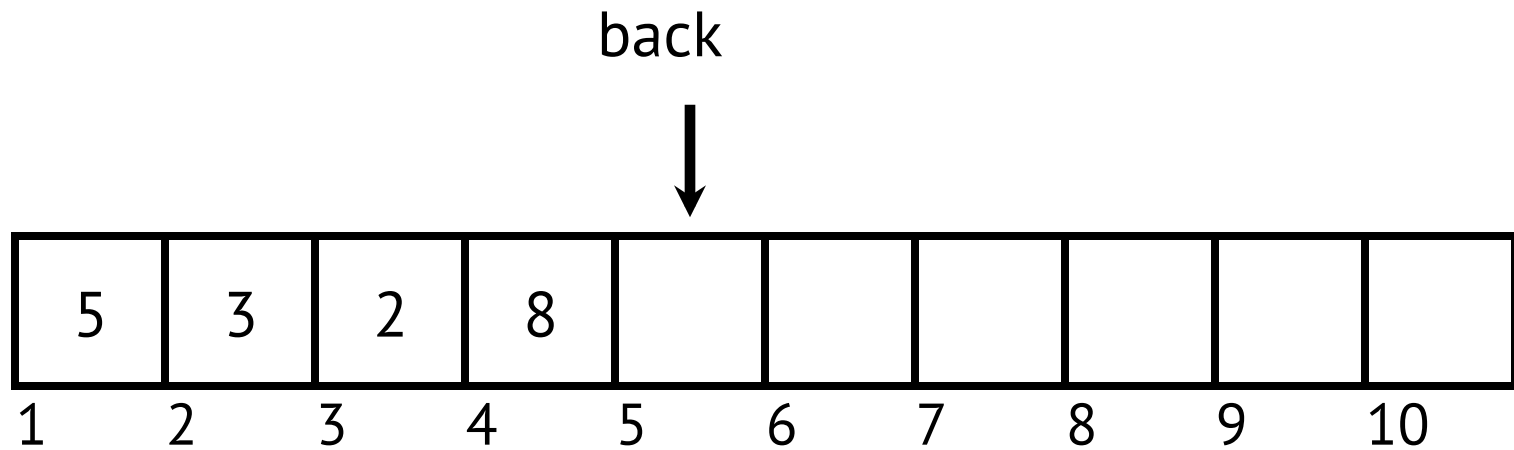
# Beispiel

Enq(8)

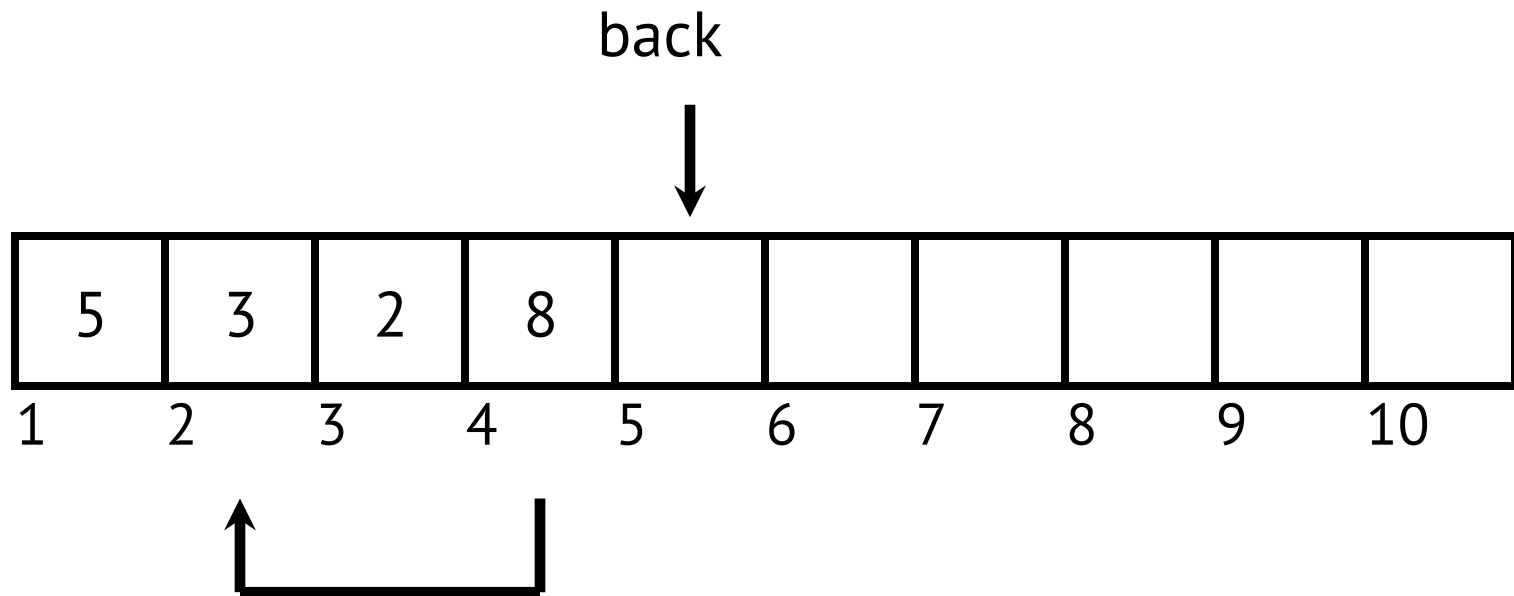
back



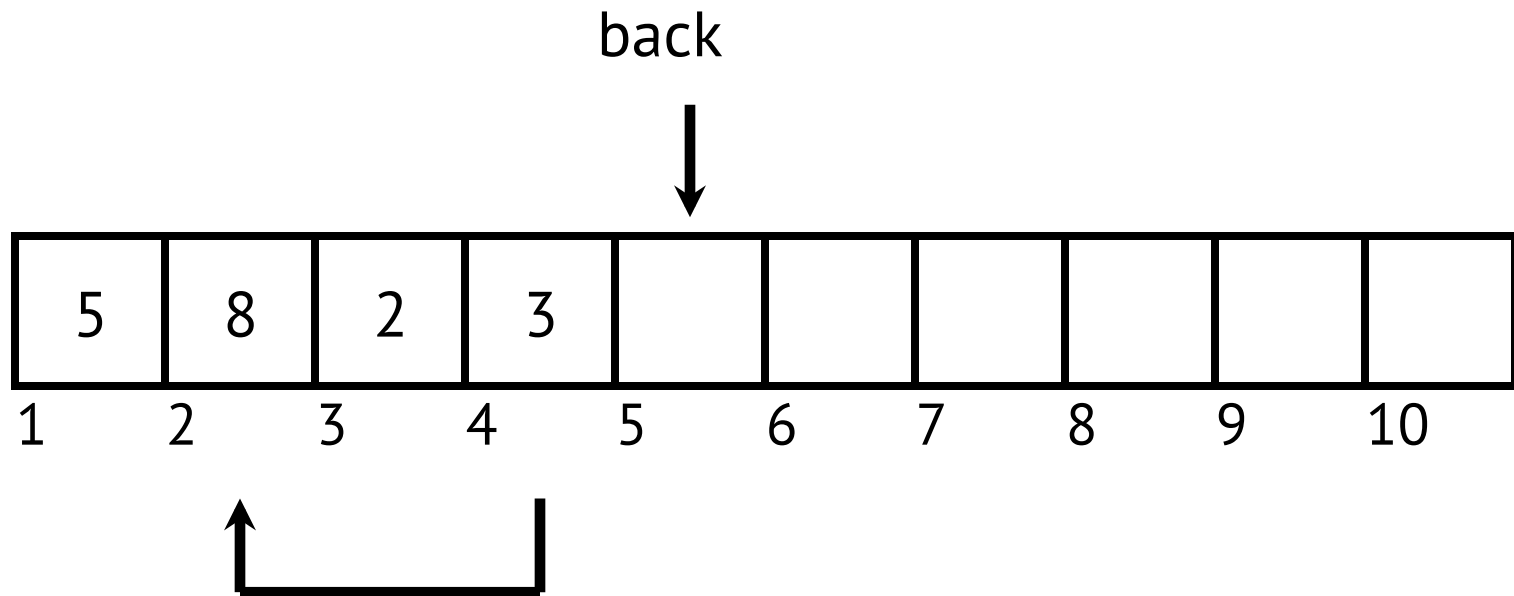
# Beispiel



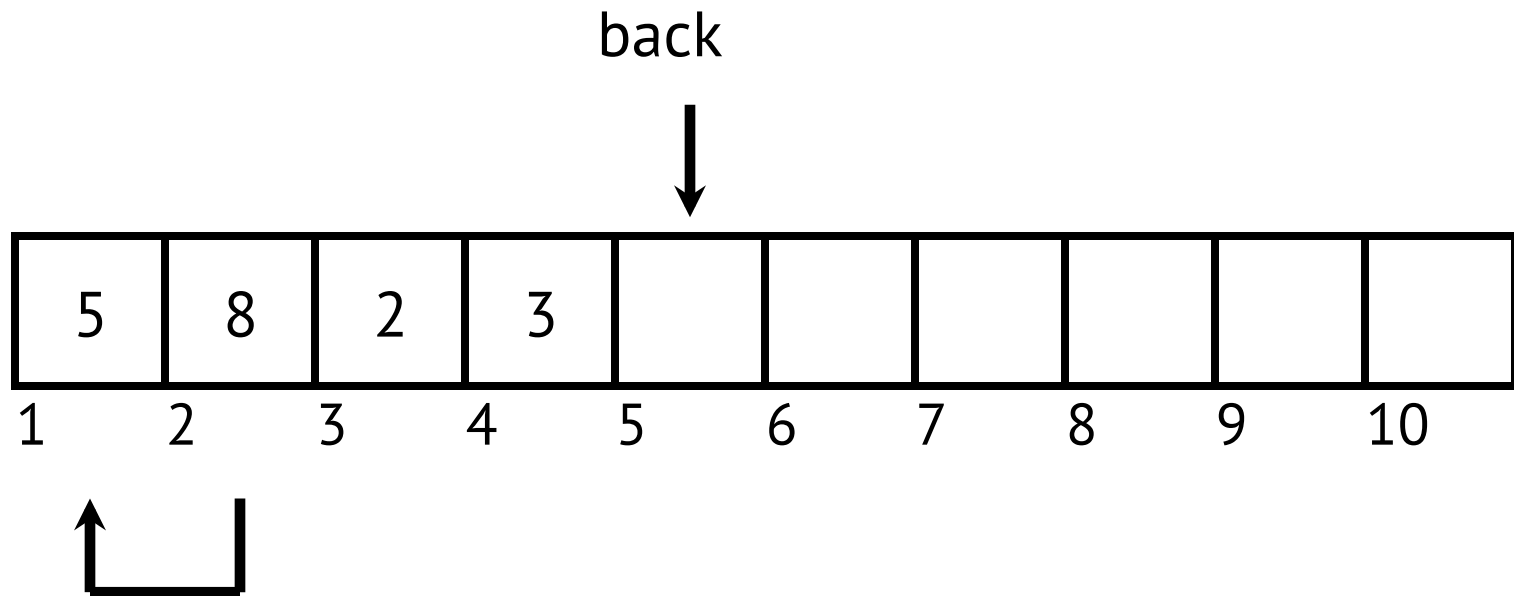
# Beispiel



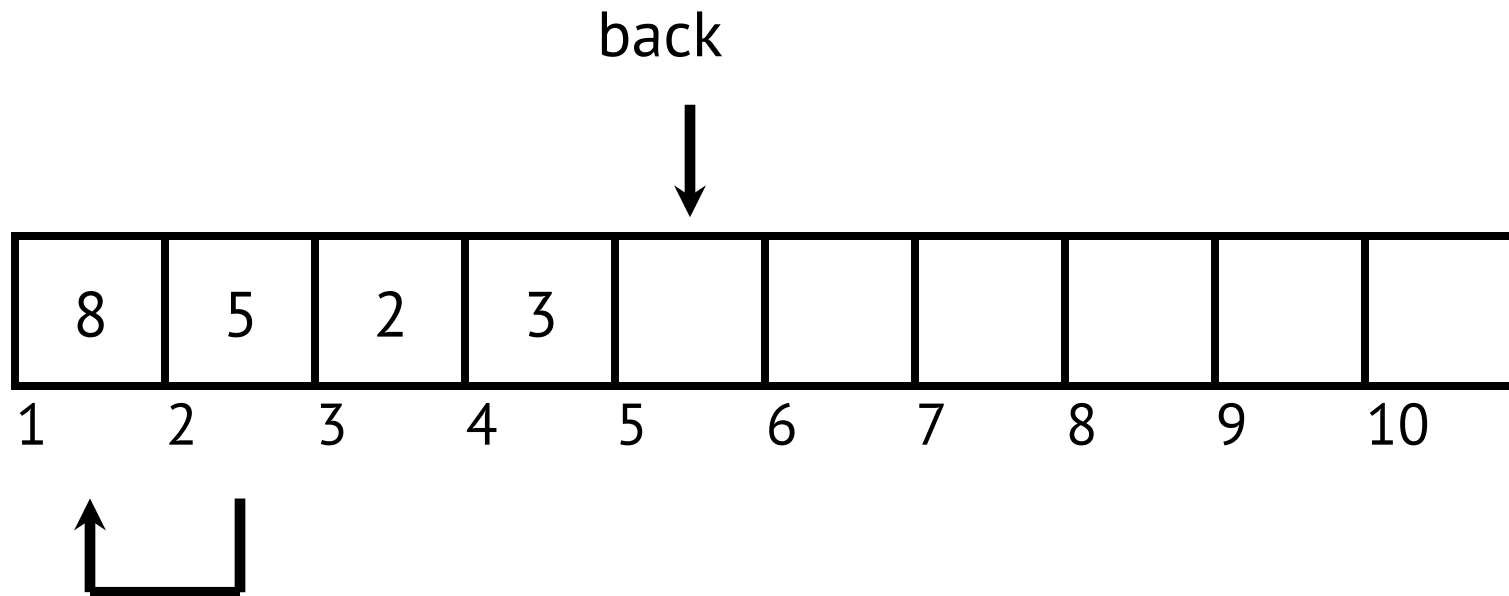
# Beispiel



# Beispiel



# Beispiel

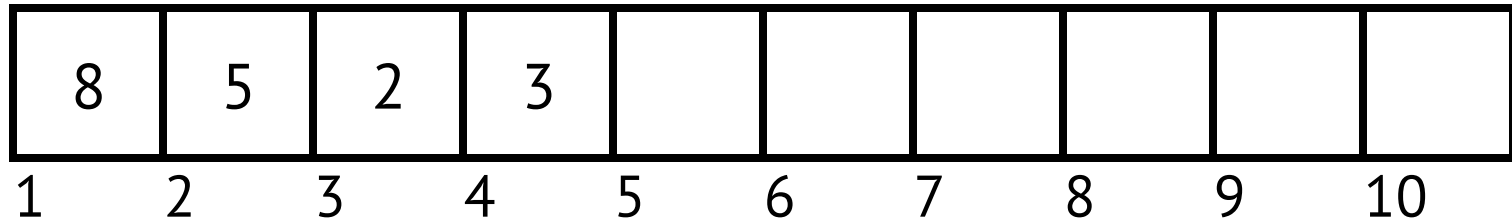




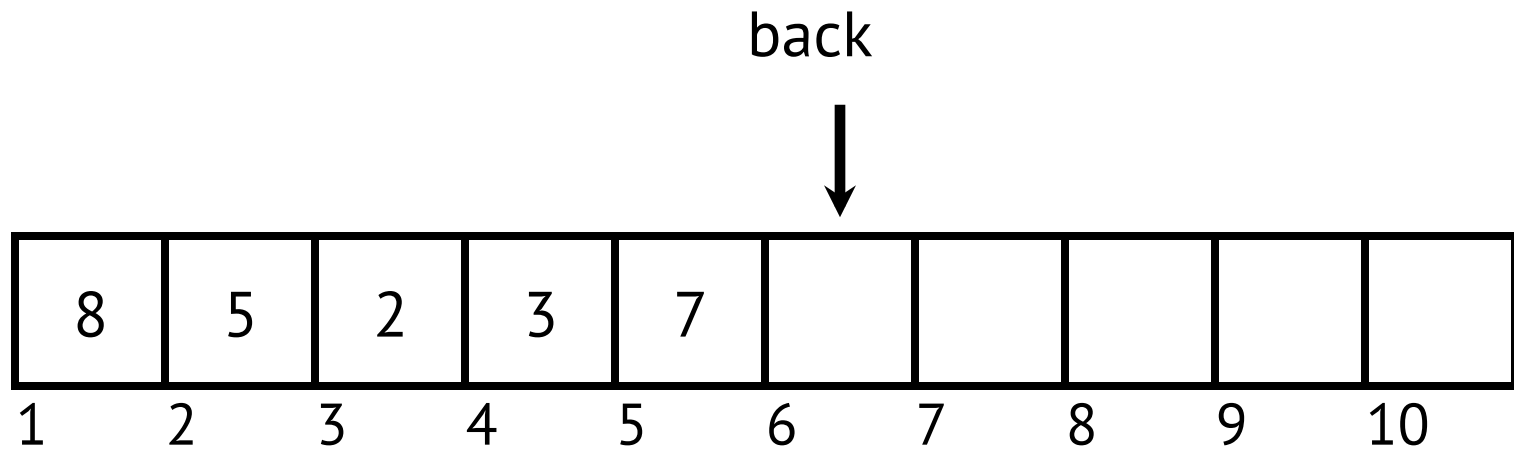
# Beispiel

Enq(7)

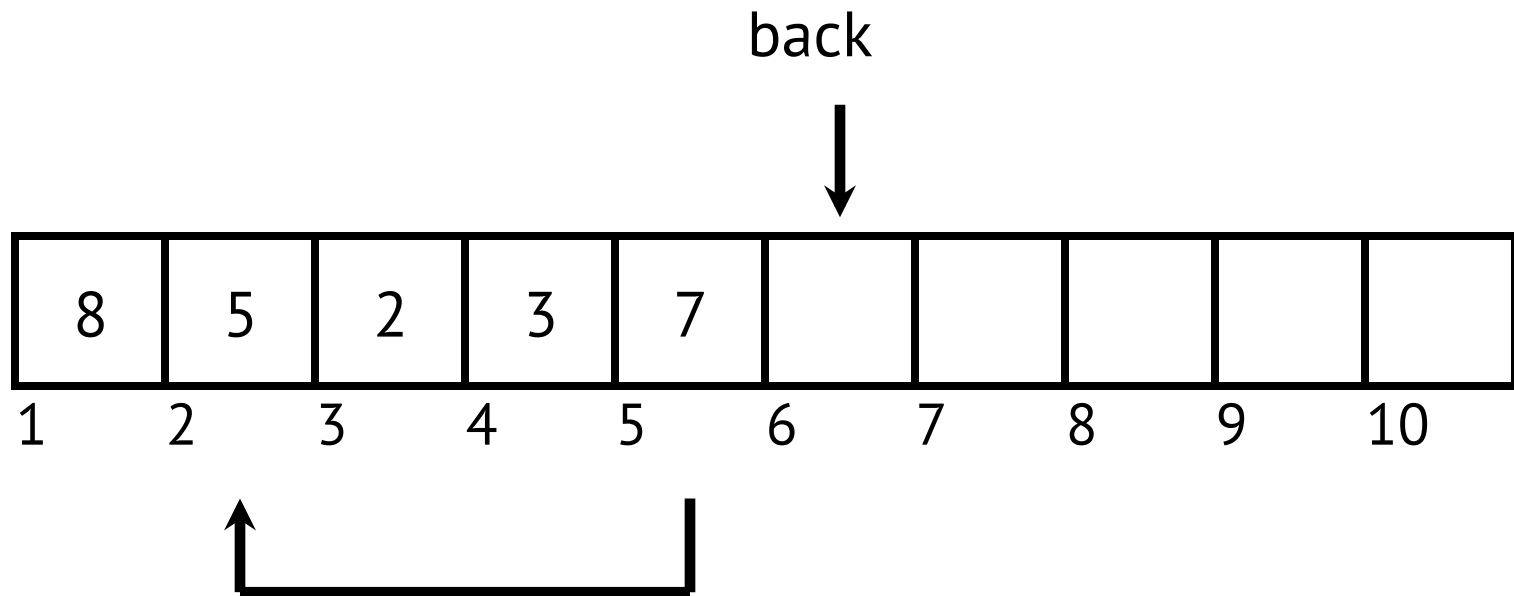
back



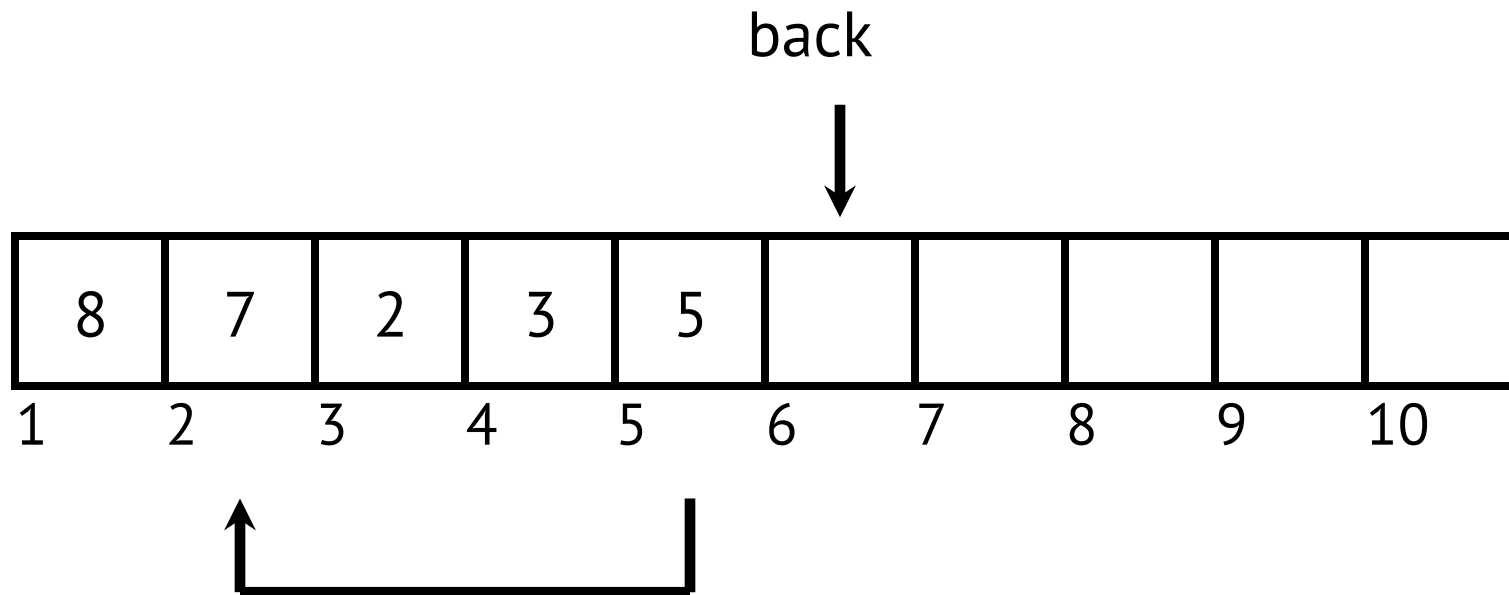
# Beispiel



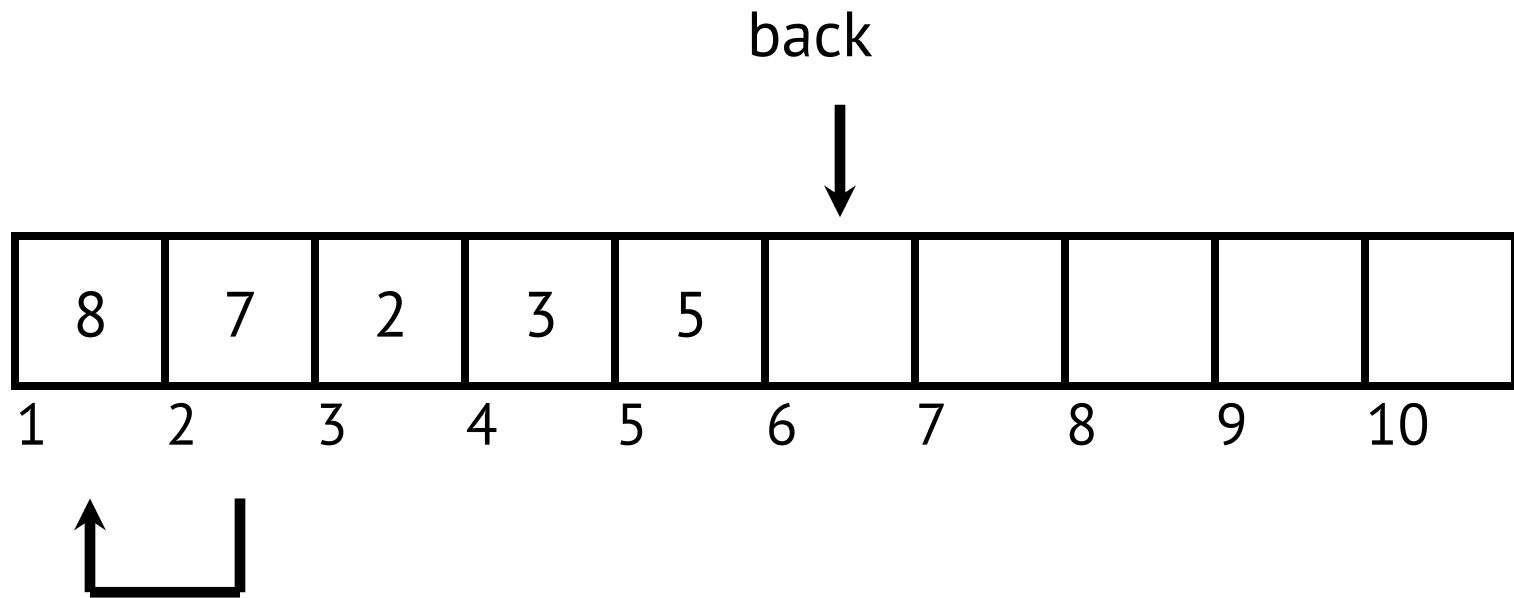
# Beispiel



# Beispiel



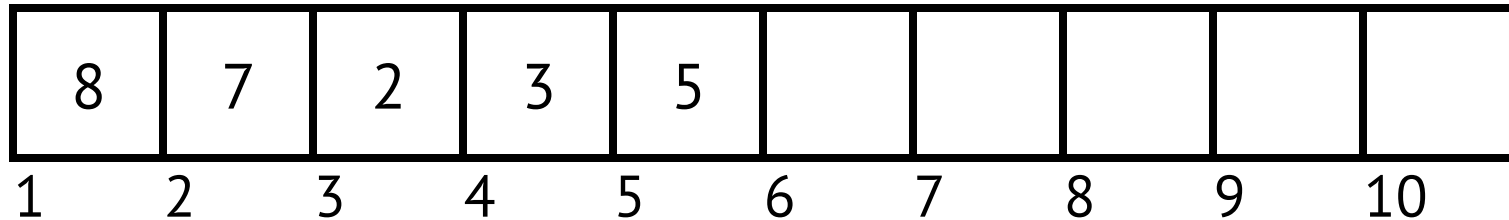
# Beispiel



# Beispiel

Enq(6)

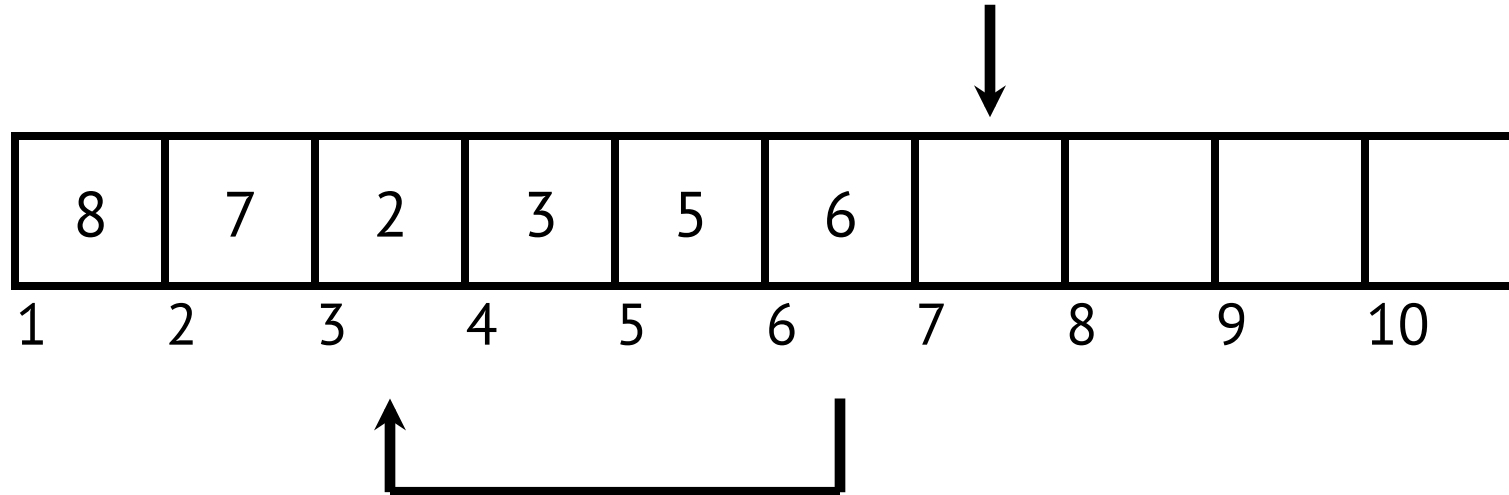
back



# Beispiel

Enq(6)

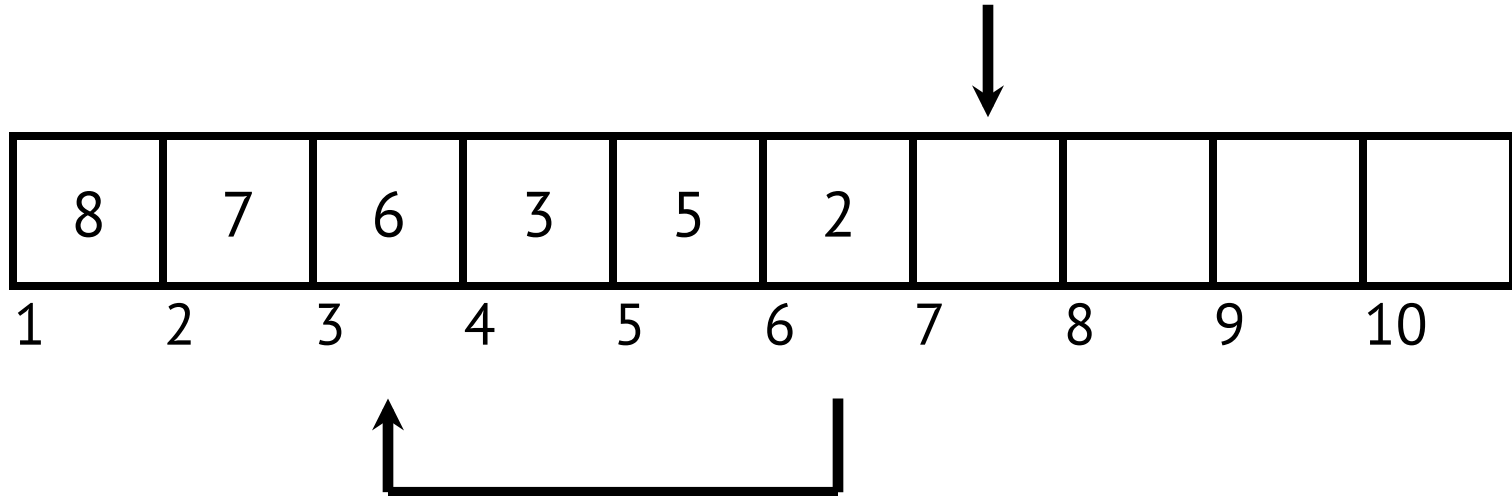
back



# Beispiel

Enq(6)

back

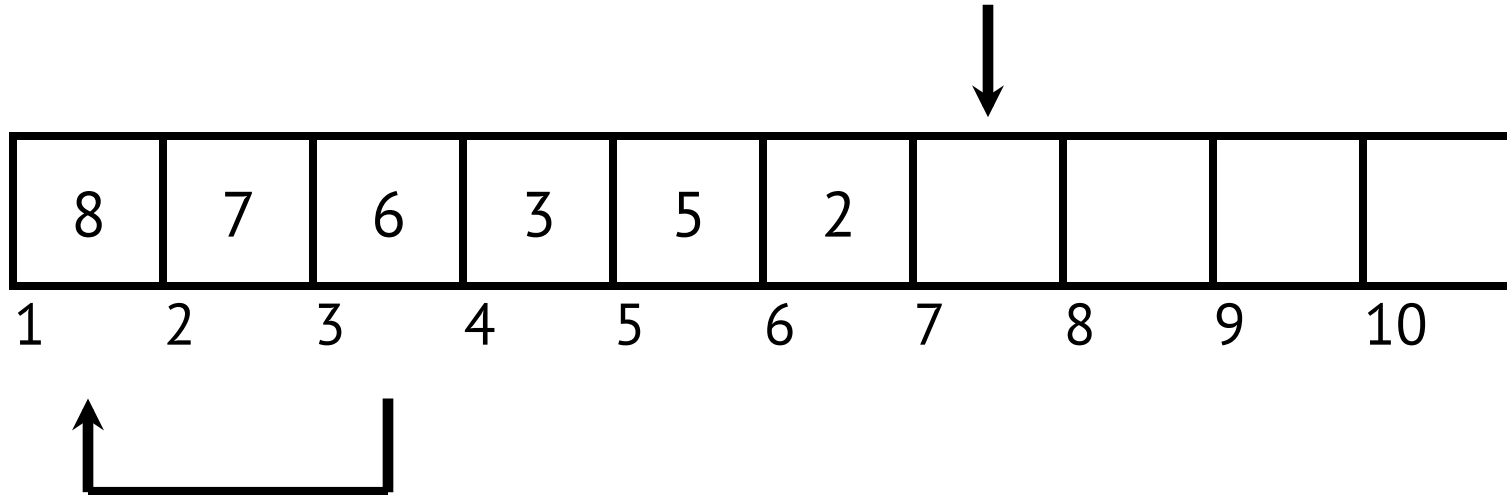




# Beispiel

Enq(6)

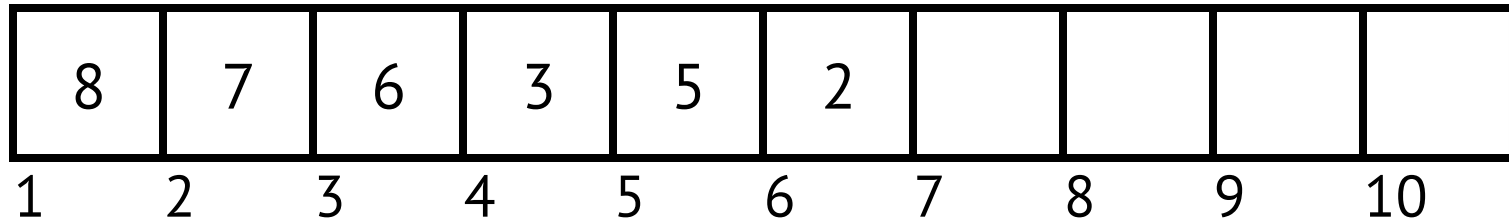
back



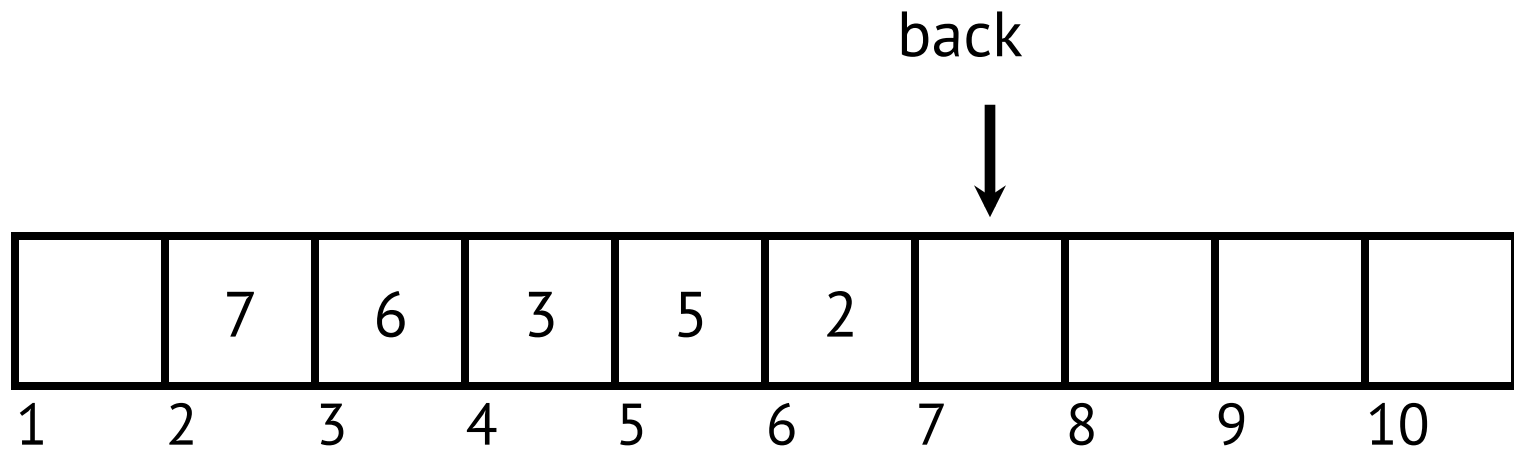
# Beispiel

Deq()

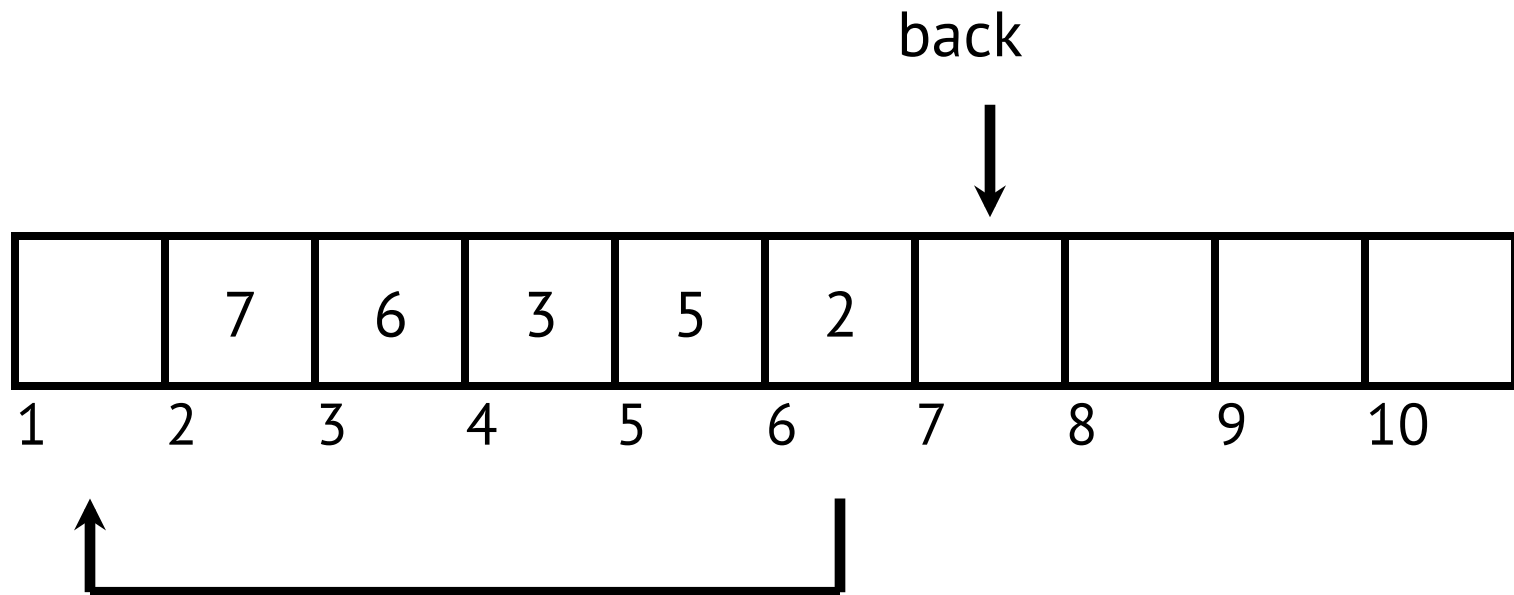
back



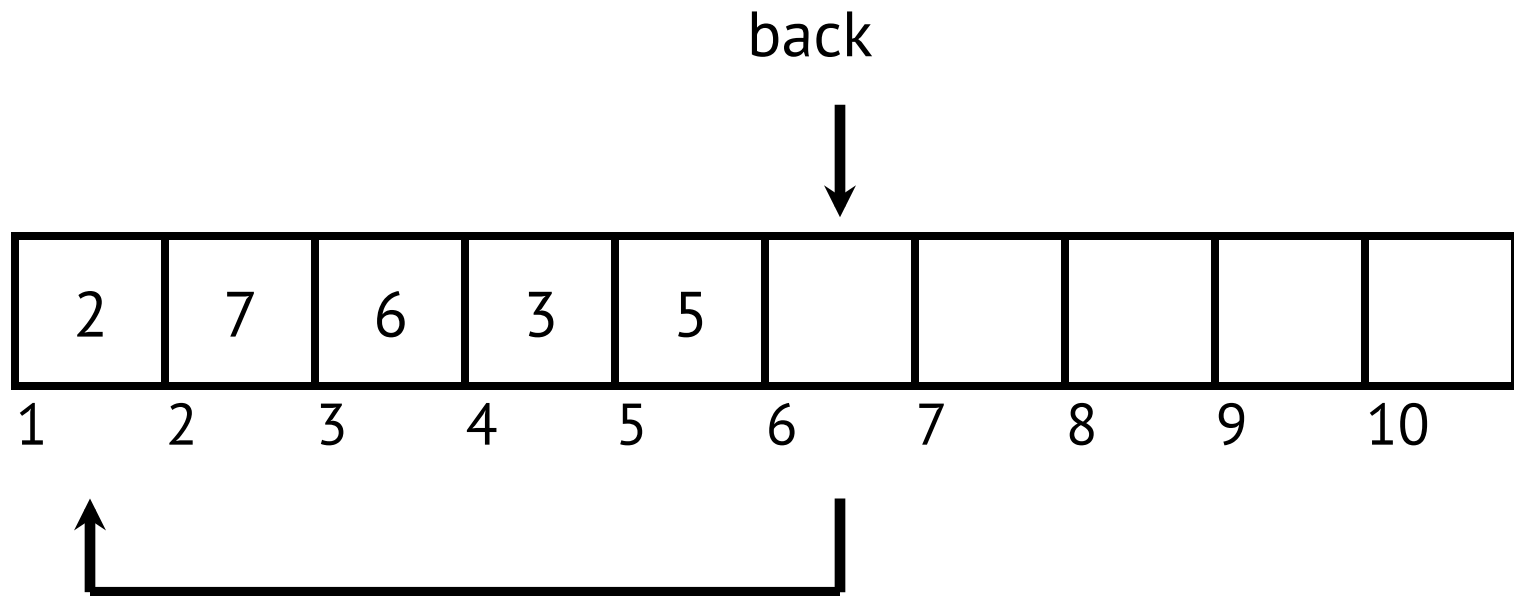
# Beispiel



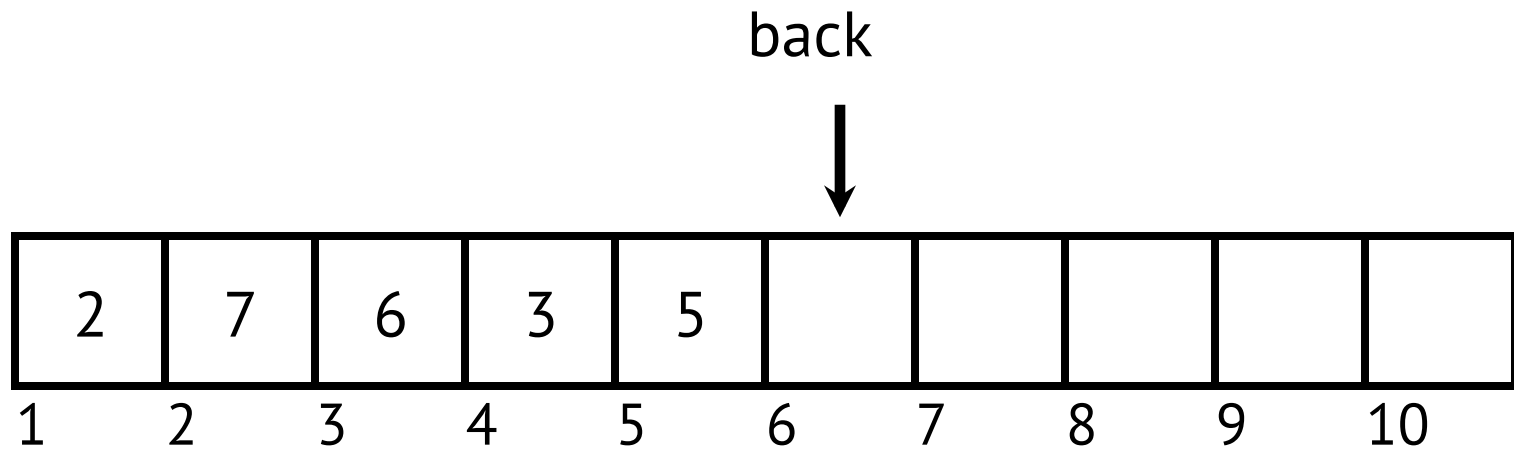
# Beispiel



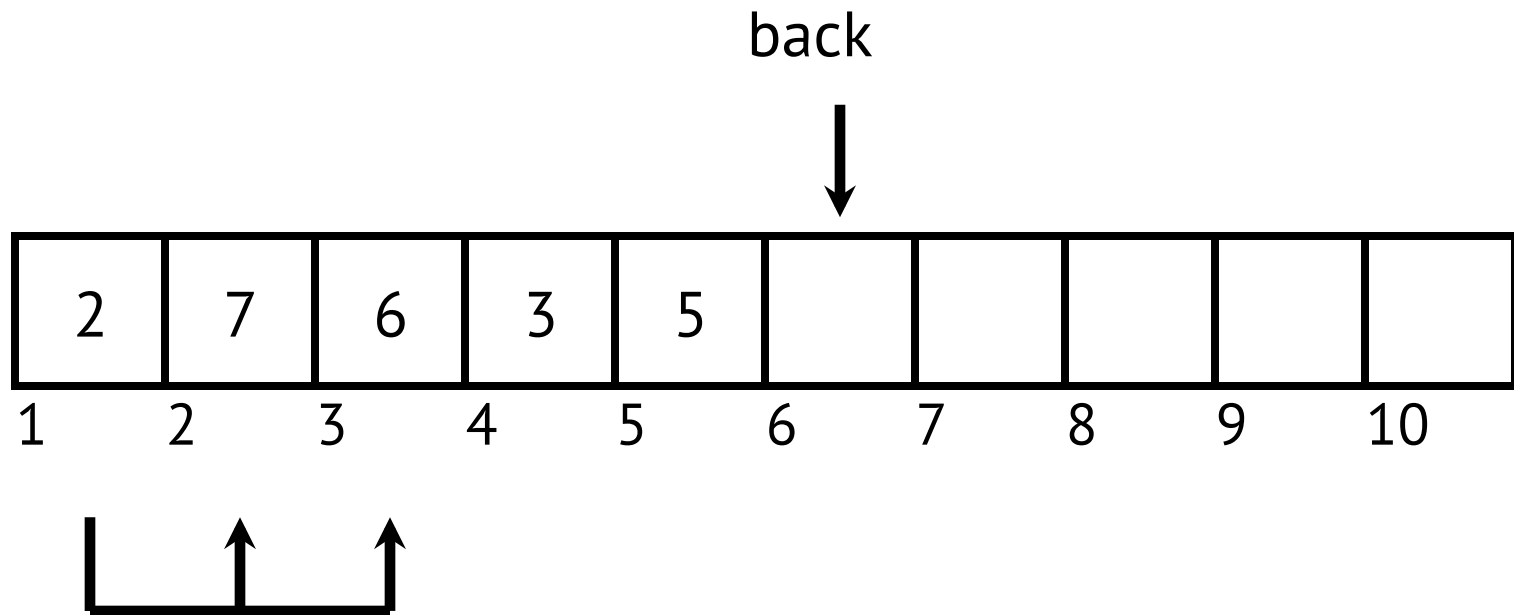
# Beispiel



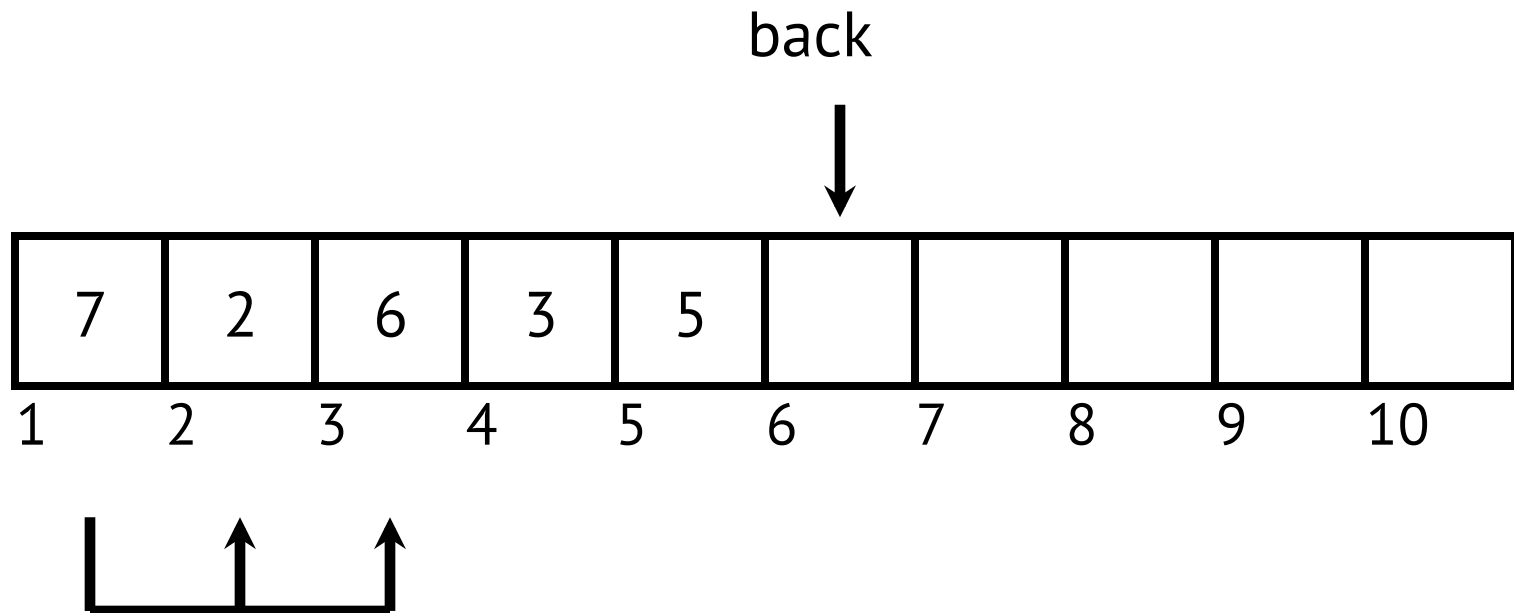
# Beispiel



# Beispiel

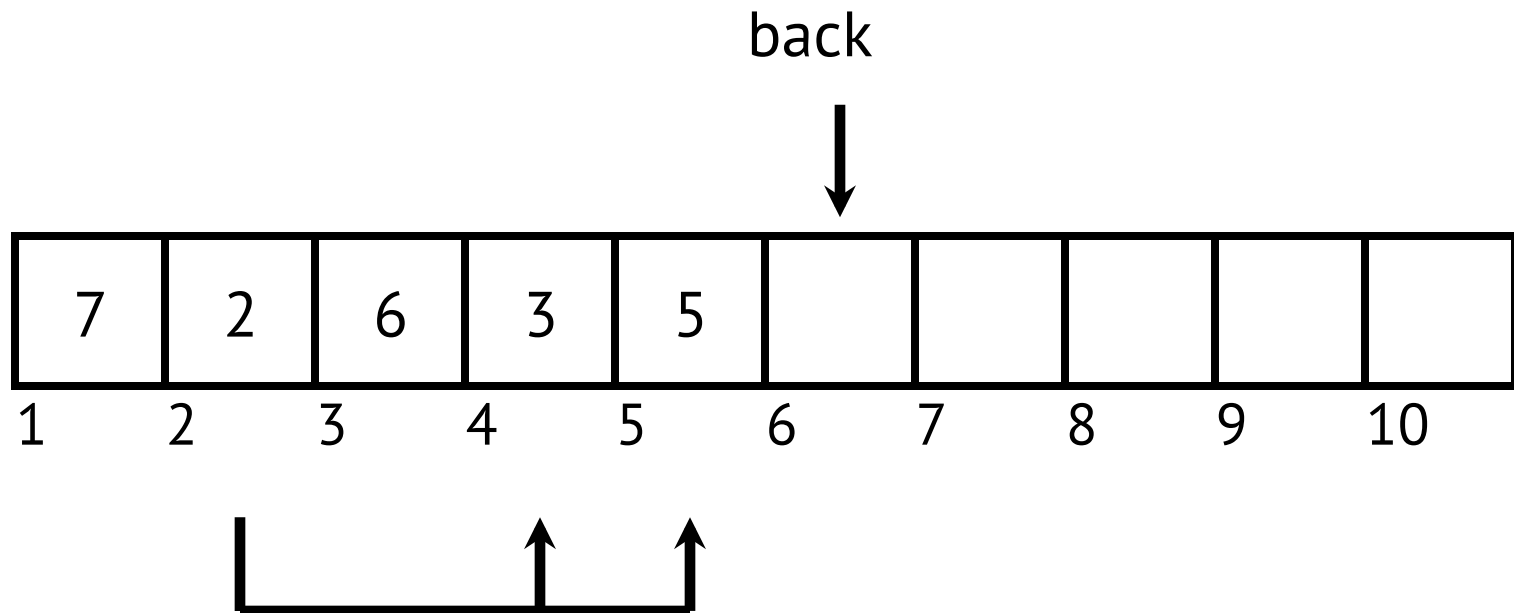


# Beispiel

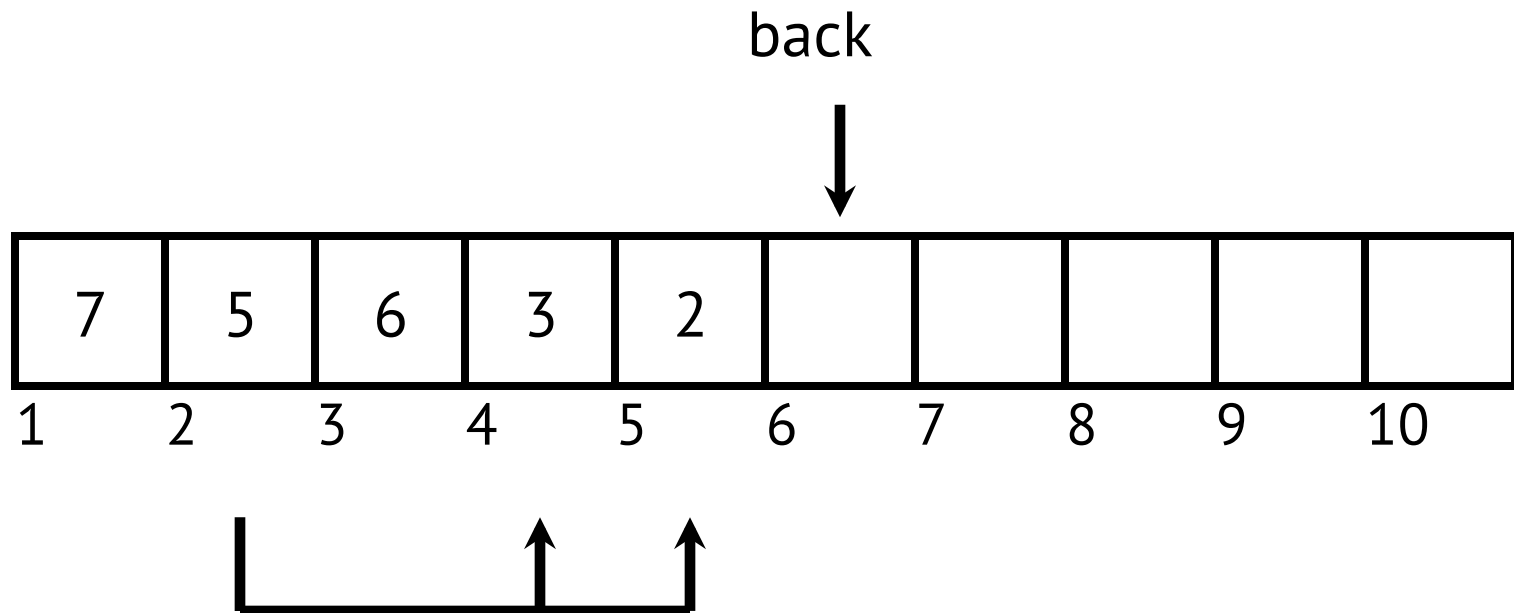




# Beispiel



# Beispiel



# Beispiel

