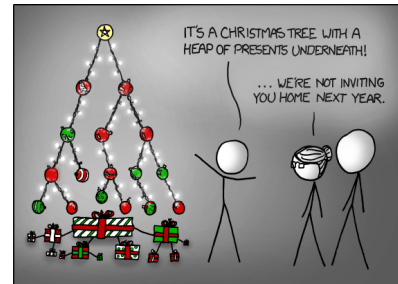


AULA 5

PQ de máximo



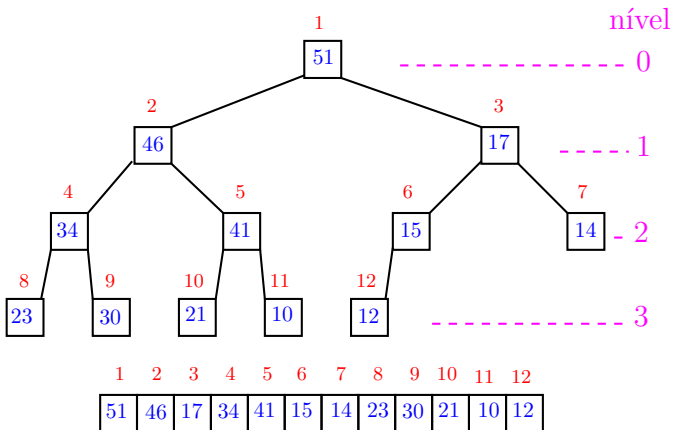
Fonte: <http://xkcd.com/835/>

Filas priorizadas, PF, Priority queues, S&W

Navigation icons

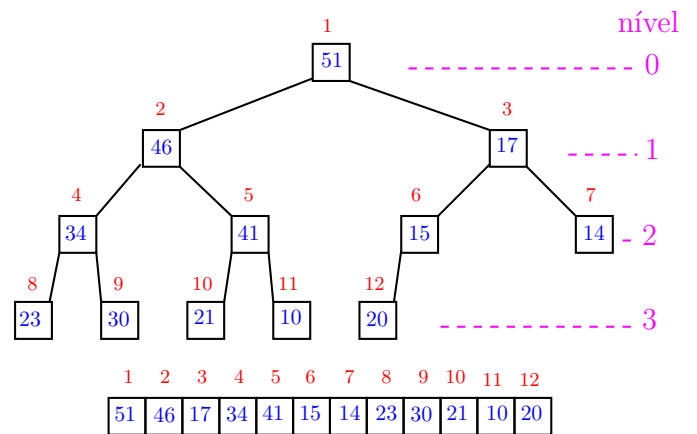
Navigation icons

max-heap



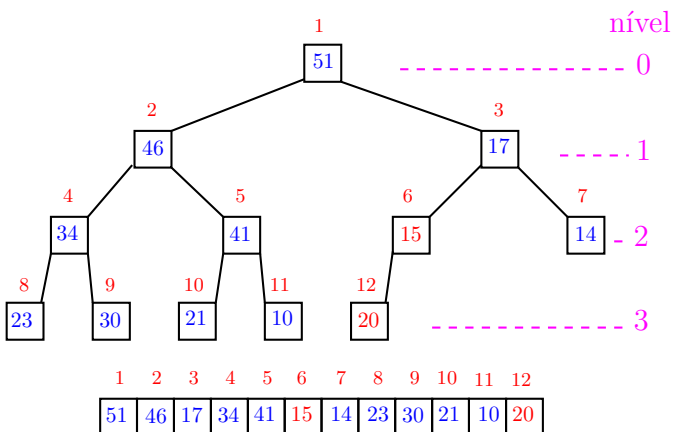
Navigation icons

swim()



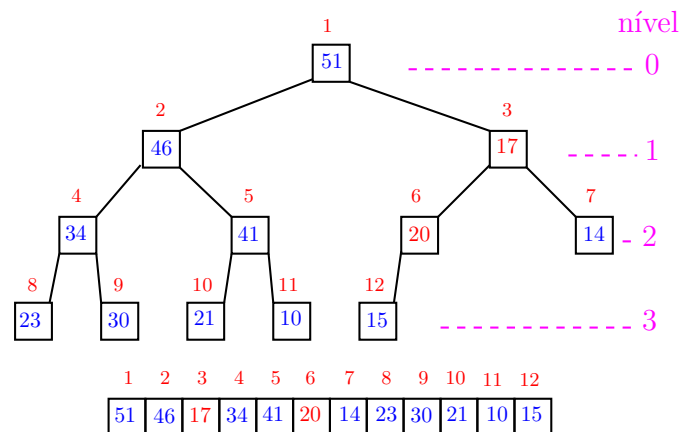
Navigation icons

swim()

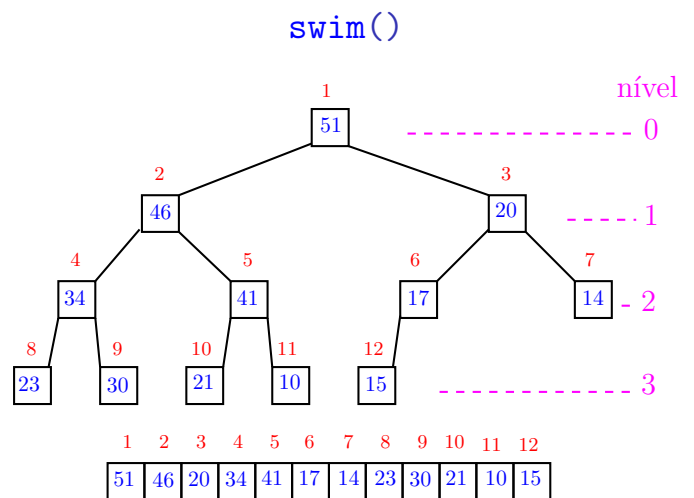
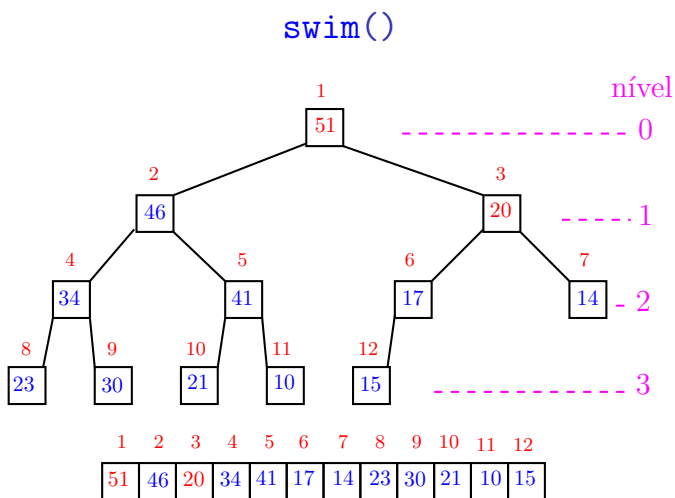


Navigation icons

swim()



Navigation icons



Função swim()

Função que recebe um **max-heap** $a[1..m-1]$ e rearranja o vetor $a[1..m]$ de modo que seja um **max-heap**.

```
public static
void swim (int f, Comparable a[]){
1  int p = f/2; Item x;
2  while (p > 1 && less(a[p],a[f])) {
3      x = a[p]; a[p] = a[f]; a[f] = x;
4      f = p; p = f/2;
  }
}
```

API PQ-máximo

```
public class MaxPQ<Item extends
    Comparable<Item>>

public class MaxPQ

MaxPQ(int cap) cria uma PQ
void insert(Item v) insere item v nesta l
Item max() devolve um máximo
Item delMax() remove e devolve
boolean isEmpty() PQ está vazia?
int size() número de itens
```

Filas priorizadas

Uma **fila priorizada** (ou **fila com prioridades**) é um ADT (*abstract data type*) que generaliza tanto a fila quanto a pilha.

Uma fila priorizada decrescente ou **PQ de máximo** é um ADT que manipula um conjunto de itens por meio de duas operações fundamentais:

- ▶ **inserção** de um novo item no conjunto e
- ▶ **remoção** de um item máximo.

Isso significa que uma fila priorizada **manipula itens comparáveis**.

PQ com itens mutáveis

Não sei se **PQ com itens mutáveis** é um bom nome para o que S&W chamam de *index priority queues*.

Em algumas aplicações é razoável permitirmos que o cliente **altere a prioridade** de um item que já está na fila.

Uma maneira de lidar com isso é **associar um único índice a cada item**.

Já comentamos essa estratégia quando tratamos de **union-find**.

API PQ-máximo mutável

```
public class IndexMaxPQ<Item> extends  
    Comparable<Item>>
```

```
public class IndexMaxPQ
```

```
    IndexMaxPQ(int maxN)  
void    insert(int k, Item item)   insere  
void    change(int k, Item item)   muda item  
boolean contains(int k)            k está associado?  
void    delete(int k)             remove k e o item k  
Item    maxM()                    retorna o menor item  
int     maxIndex()                retorna o índice do  
int     delMax()                  retorna o menor item  
boolean isEmpty()                 está vazia?  
int     size()                     número de itens
```

◀ ▶ ⏪ ⏩ 🔍 ↻