Heap Esquerdista

Estrutura de Dados II

Prof Jairo Francisco de Souza

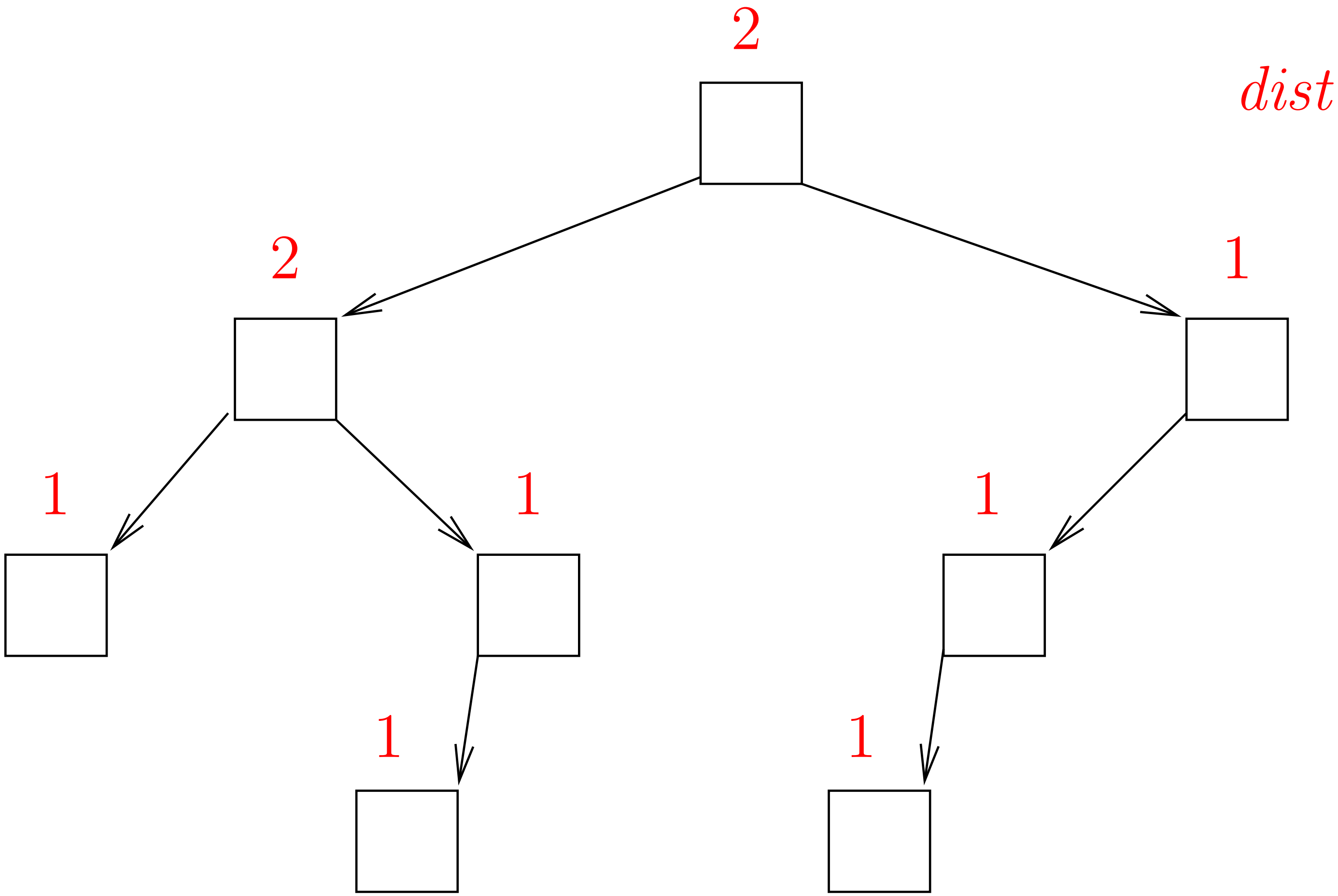
Árvores esquerdistas

Cada nó x tem quatro campos:

1. $esq[x]$: filho esquerdo de x;
2. $dir[x]$: filho direito de x;
3. $dist[x]$: menor comprimento de um caminho de x a NIL.

DIST (x)
1 **se** $x = \text{NIL}$
2 **então devolva** 0
3 **senão devolva** $1 + \min\{\text{DIST}(esq[x]), \text{DIST}(dir[x])\}$

Exemplo



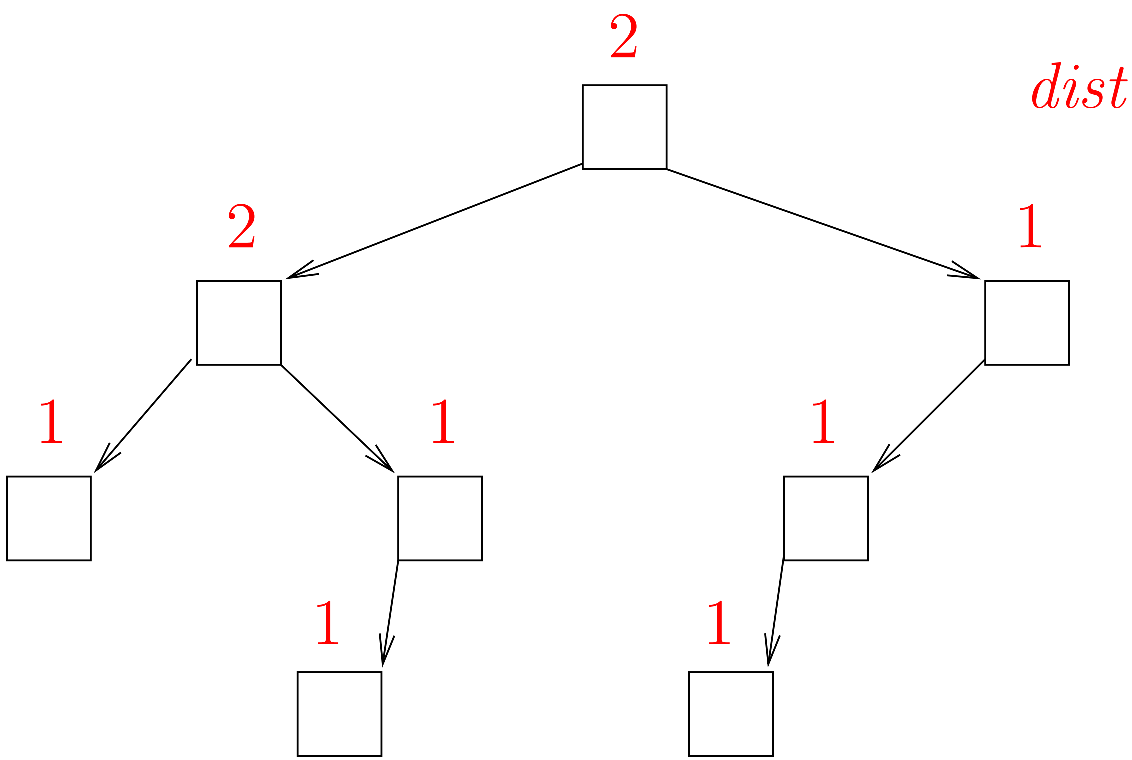
Árvores esquerdistas

Uma árvore é esquerdista se

$$dist[esq[x]] \geq dist[dir[x]]$$

para todo nó x ($dist[\text{NIL}] = 0$).

Exemplo 1:



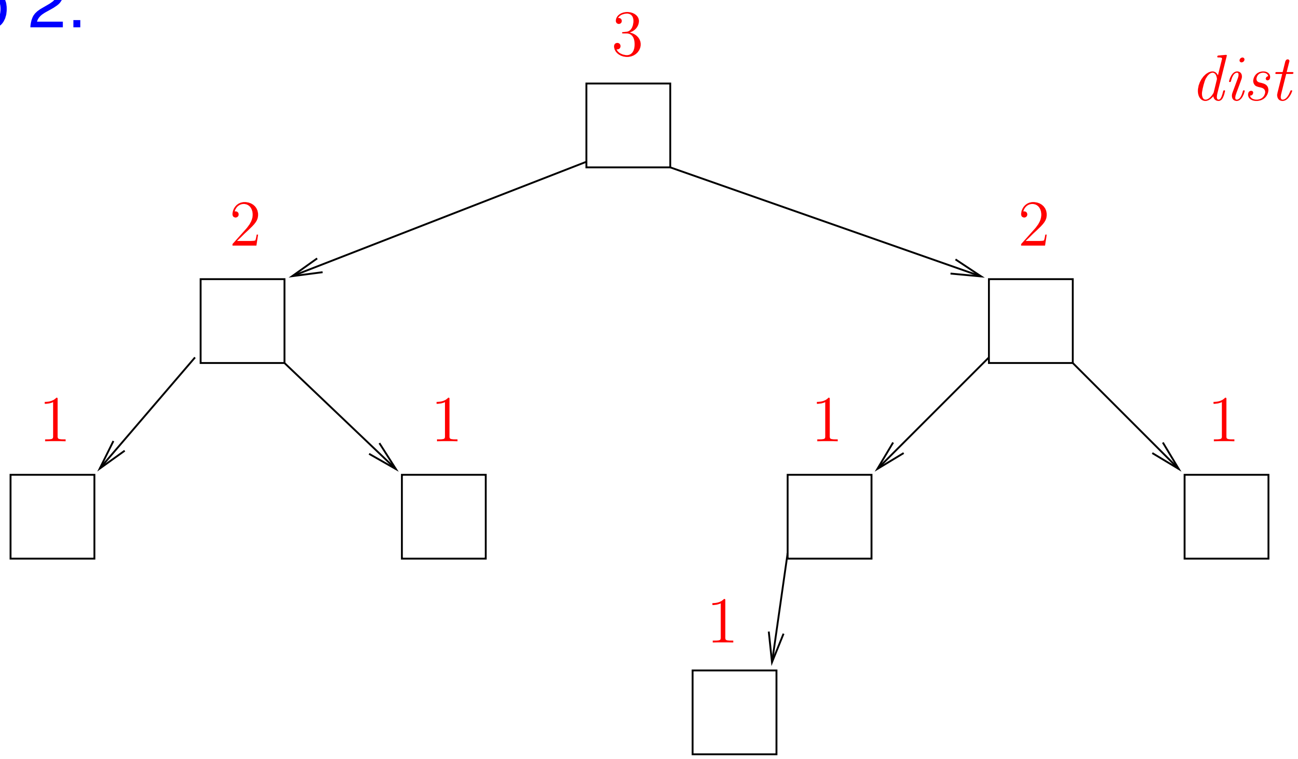
Árvores esquerdistas

Uma árvore é esquerdista se

$$dist[esq[x]] \geq dist[dir[x]]$$

para todo nó x ($dist[\text{NIL}] = 0$).

Exemplo 2:



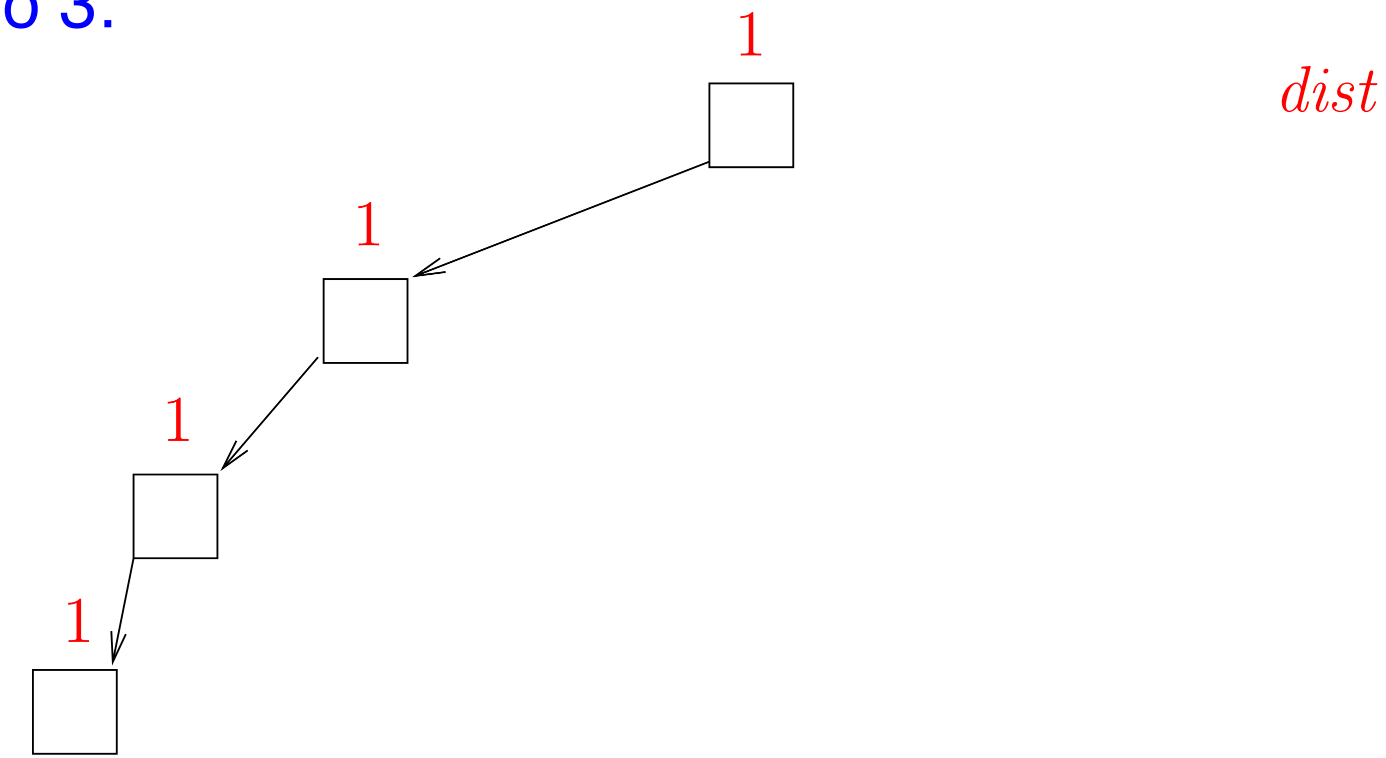
Árvores esquerdistas

Uma árvore é esquerdista se

$$dist[esq[x]] \geq dist[dir[x]]$$

para todo nó x ($dist[\text{NIL}] = 0$).

Exemplo 3:



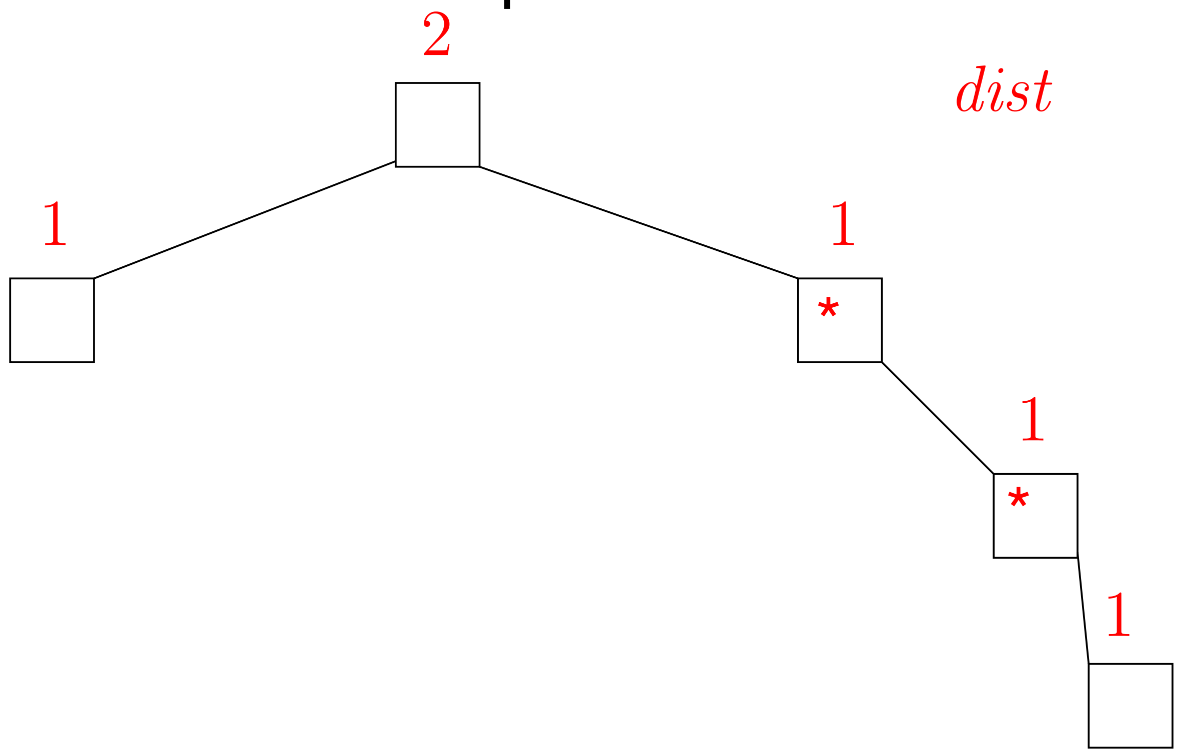
Árvores esquerdistas

Uma árvore é esquerdista se

$$dist[esq[x]] \geq dist[dir[x]]$$

para todo nó x ($dist[\text{NIL}] = 0$).

Exemplo 4: árvore não-esquerdista



Caminho direitista

O caminho direitista de um nó x é a seqüência

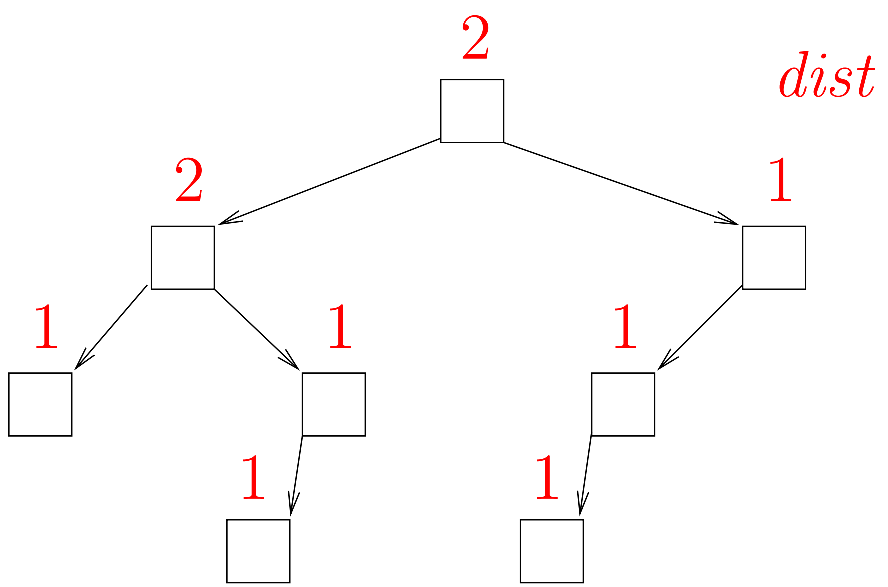
$$\langle x, dir[x], dir[dir[x]], \ldots, \text{NIL} \rangle.$$

$dcomp[x]$:= número de nós no caminho direitista de x

$tam[x]$:= número de nós na árvore de raiz x

Se x é um nó de uma árvore esquerdista, então

$$dist[x] = dcomp[x].$$



Fato importante

$tam[x] :=$ número de nós na árvore de raiz x

Se x é um nó de uma árvore esquerdista, então

$$tam[x] \geq 2^{dist[x]} - 1.$$

Fato importante

$tam[x] :=$ número de nós na árvore de raiz x

Se x é um nó de uma árvore esquerdista, então

$$tam[x] \geq 2^{dist[x]} - 1.$$

Prova: Seja $d := dist[x]$.

Se $d = 1$, então $tam[x] \geq 1 = 2^d - 1$.

Suponha que $d \geq 2$ e que a desigualdade vale para $d - 1$.

Temos que $dist[dir[x]] = d - 1$ e que existe um nó y na árvore de raiz $esq[x]$ tal que $dist[y] = d - 1$.

Fato importante

$tam[x]$:= número de nós na árvore de raiz x

Fato 2. Se x é um nó de uma árvore esquerdista, então

$$tam[x] \geq 2^{dist[x]} - 1.$$

Prova: (continuação)
Logo,

$$\begin{aligned} tam[x] &= tam[esq[x]] + tam[dir[x]] + 1 \\ &\geq tam[y] + tam[dir[x]] + 1 \\ &\overset{\text{hi}}{\geq} 2^{d-1} - 1 + 2^{d-1} - 1 + 1 \\ &= 2^d - 1 \end{aligned}$$

Conseqüência

Se x é um nó de uma árvore esquerdista, então

$$dist[x] = dcomp[x] \leq \lfloor \lg(tam[x] + 1) \rfloor = \mathrm{O}(tam[x]).$$

Em particular:

Se x é raiz de uma árvore esquerdista com m nós,

$$dist[x] = dcomp[x] \leq \lfloor \lg(m + 1) \rfloor = \mathrm{O}(\lg m).$$

Prova: $m \geq 2^d - 1 \Rightarrow m + 1 \geq 2^d \Rightarrow \lfloor \lg(m + 1) \rfloor \geq d$.

Heap esquerdista

H := árvore

$raiz[H]$:= raiz de H

$prior[x]$:= prioridade do nó x

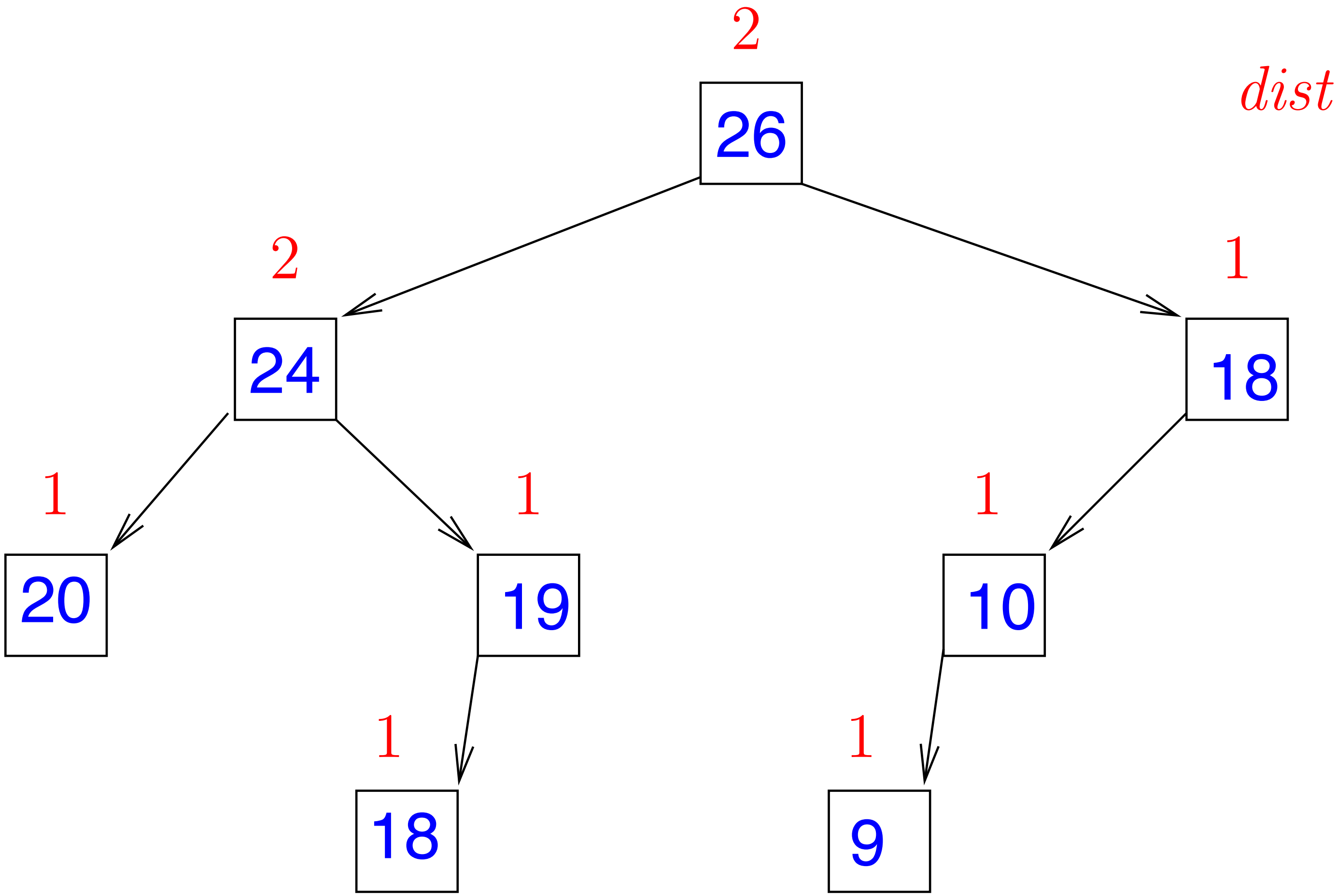
$pai[x]$:= pai do nó x

Um heap esquerdista H é uma árvore esquerdista que satisfaz

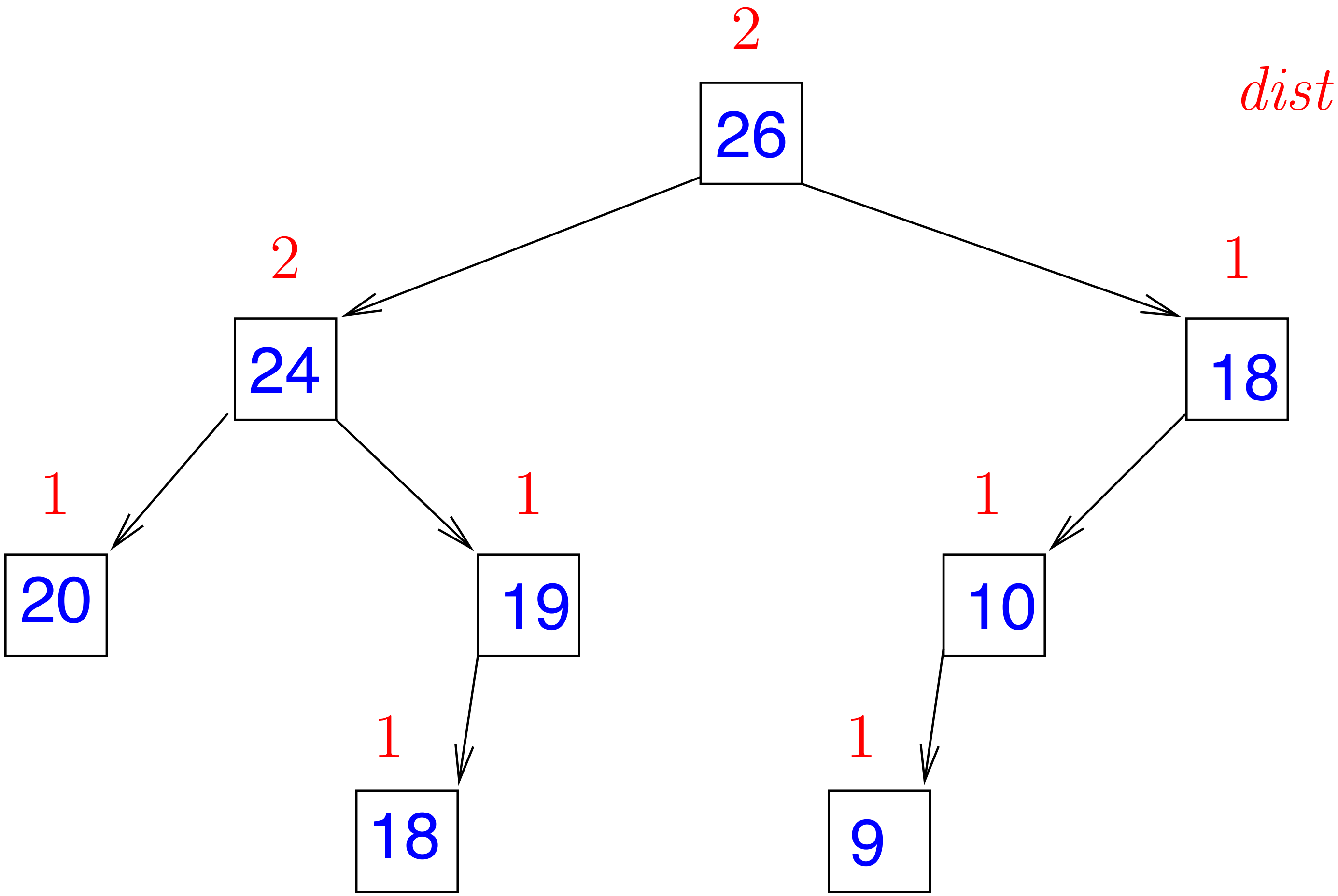
$$prior[pai[x]] \geq prior[x]$$

para todo nó $x \neq raiz[H]$.

Heap esquerdista

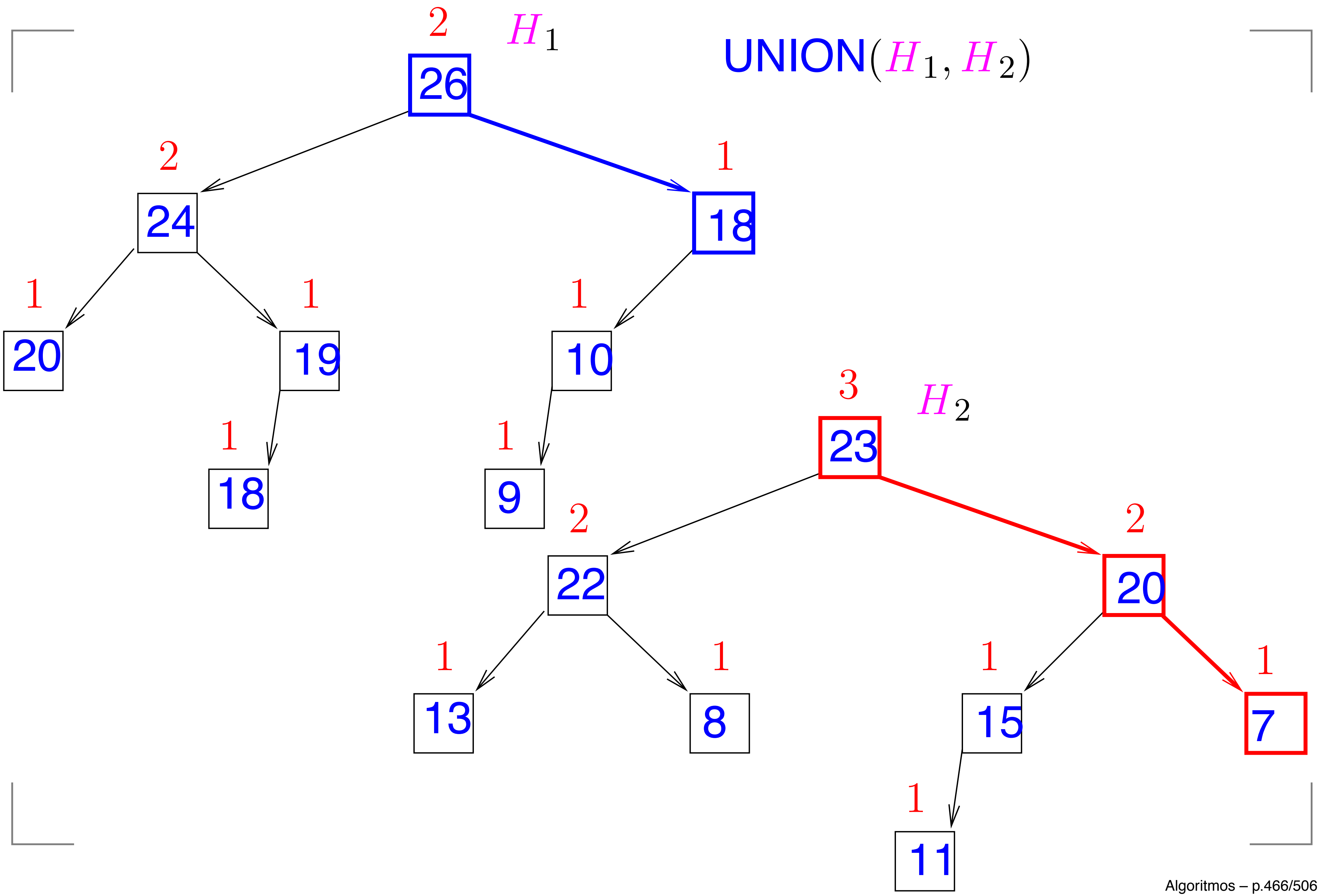


Heap esquerdista

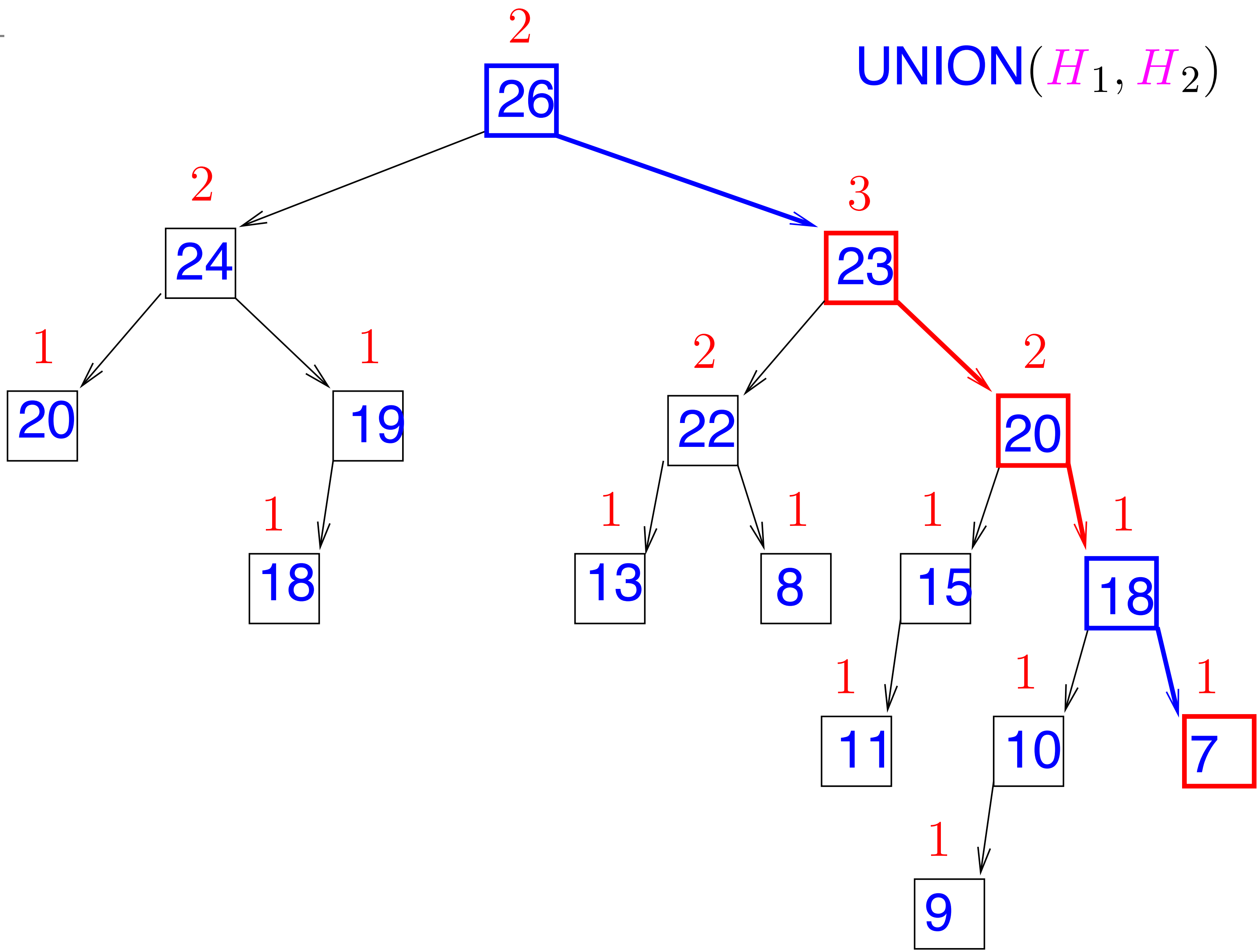


Rotina básica de manipulação

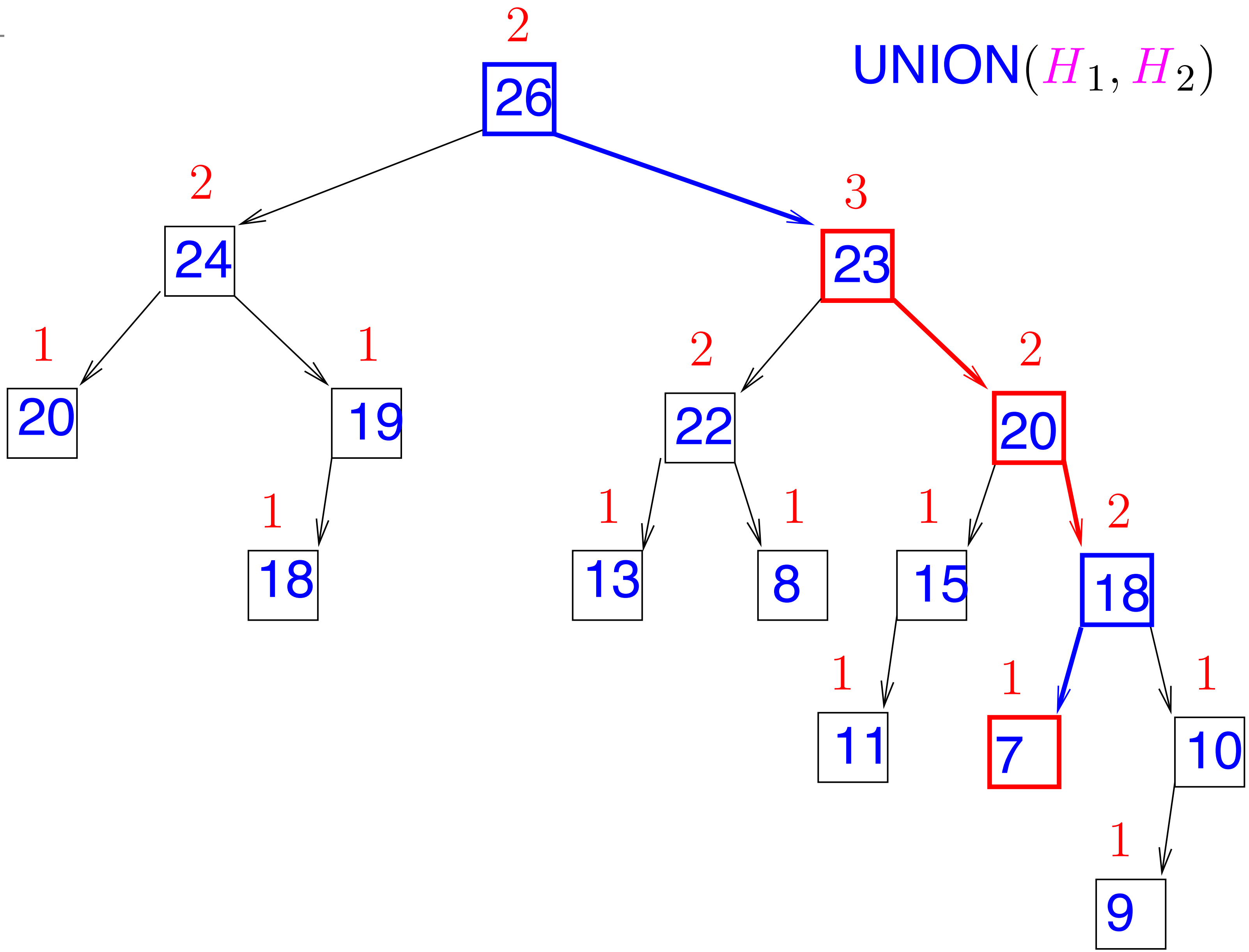
UNION(H_1, H_2)



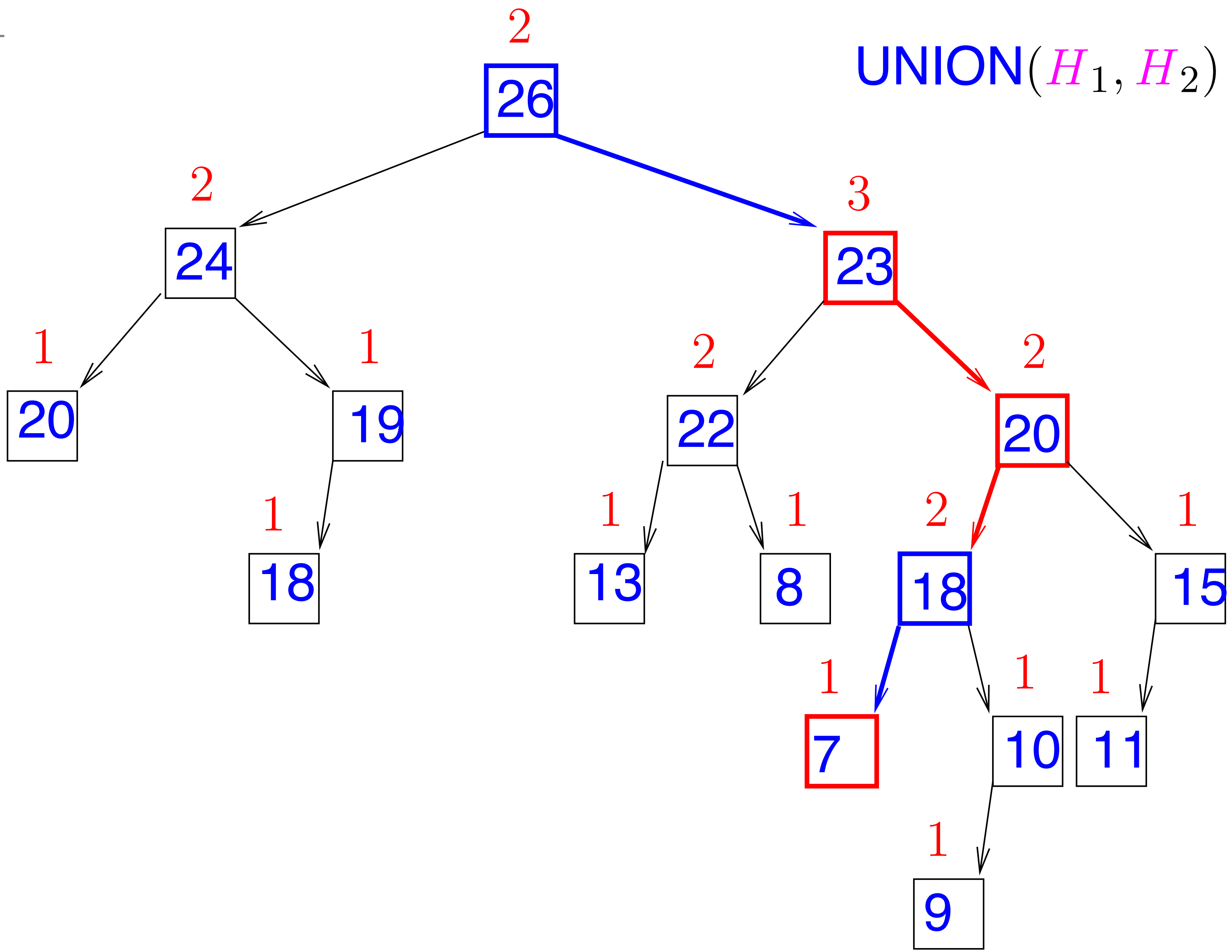
Rotina básica de manipulação



Rotina básica de manipulação

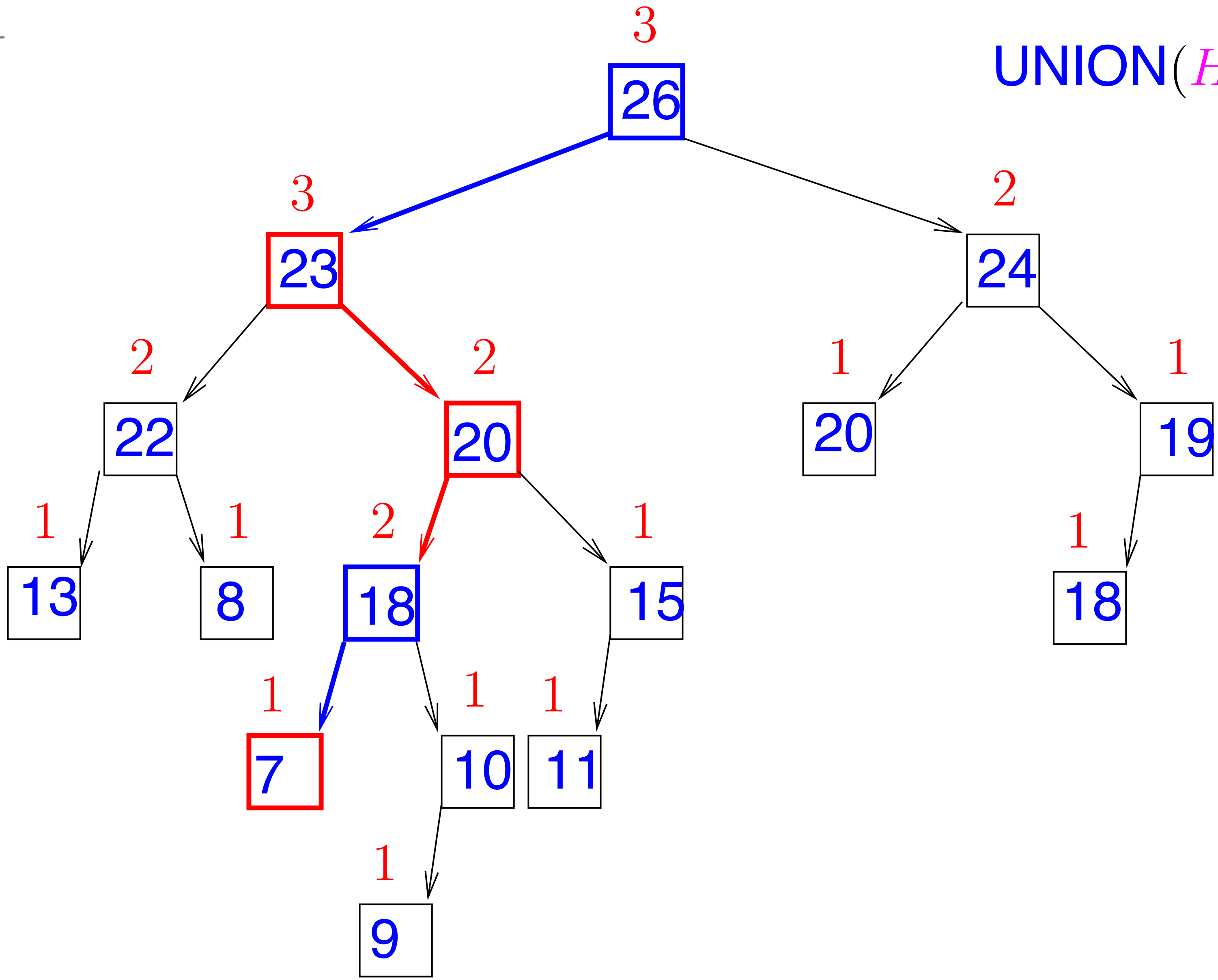


Rotina básica de manipulação



Rotina básica de manipulação

UNION(H_1, H_2)



Rotina básica de manipulação

LEFTIST-HEAP-UNION (H_1, H_2)

se $raiz[H_1] = \text{NIL}$ **então devolva** H_2
se $raiz[H_2] = \text{NIL}$ **então devolva** H_1
se $prior[raiz[H_1]] < prior[raiz[H_2]]$ **então** $H_1 \leftrightarrow H_2$
$x_1 \leftarrow raiz[H_1] \quad x_2 \leftarrow raiz[H_2]$
se $esq[x_1] = \text{NIL}$ **então** $esq[x_1] \leftarrow x_2$
senão $H' \leftarrow$ MAKE-LEFTIST-HEAP$()$
    $raiz[H'] \leftarrow dir[x_1]$
    $H' \leftarrow$ LEFTIST-HEAP-UNION(H', H_2)
    $dir[x_1] \leftarrow raiz[H']$
    se $dist[esq[x_1]] < dist[dir[x_1]]$
        então $esq[x_1] \leftrightarrow dir[x_2]$
    $dist[x_1] \leftarrow dist[dir[x_1]] + 1$
devolva H_1

Consumo de tempo

O consumo de tempo do algoritmo LEFTIST-HEAP-UNION no pior caso é proporcional a

$$dcomp(raiz[H_1]) + dcomp(raiz[H_2]) = \mathrm{O}(\lg m)$$

onde $m = tam(raiz[H_1]) + tam(raiz[H_2])$

O consumo de tempo do algoritmo LEFTIST-HEAP-UNION é $\mathrm{O}(\lg m)$.

Fila com heap esquerdista

Rotina que insere um nó x em um heap esquerdista H.

A rotina supões que $prior[x]$ já foi definido.

LEFTIST-HEAP-INSERT (H, x)
1 $H' \leftarrow$ MAKE-LEFTIST-HEAP$()$
2 $esq[x] \leftarrow \text{NIL}$
3 $dir[x] \leftarrow \text{NIL}$
4 $dist[x] \leftarrow 1$
5 $raiz[H'] \leftarrow x$
6 $H \leftarrow$ LEFTIST-HEAP-UNION(H, H')

Consome tempo $\mathrm{O}(\lg m)$.

Fila com heap esquerdista

Rotina que remove e devolve o nó de maior prioridade de um heap esquerdista H.

```
LEFTIST-HEAP-EXTRACT (H)
    x ← raiz[H]
    H′ ← MAKE-LEFTIST-HEAP()
    raiz[H′] ← esq[x]
    raiz[H] ← dir[x]
    H ← LEFTIST-HEAP-UNION(H, H′)
    devolva x
```

Consome tempo $\mathrm{O}(\lg m)$.