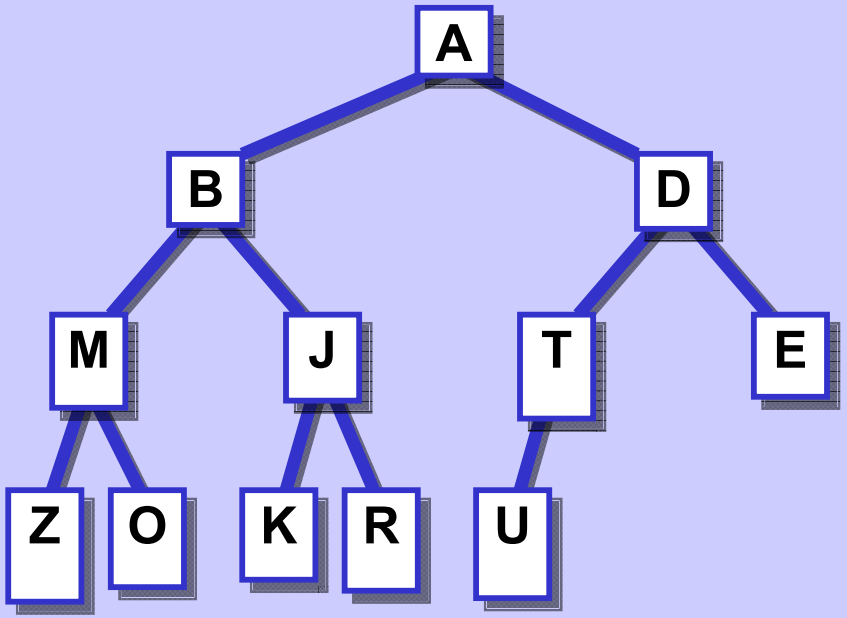
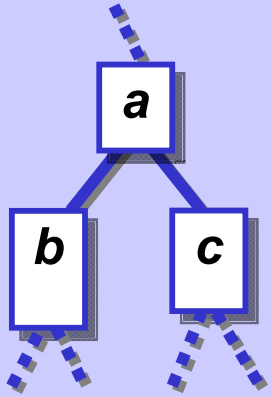
Heap sort

Halda



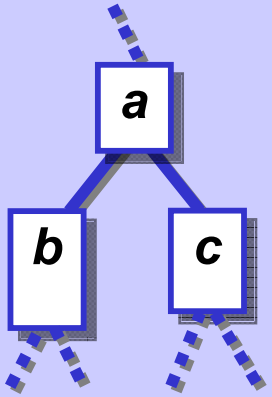
Pravidlo haldy



$$a \leq b \;\&\&\; a \leq c$$

Heap sort

Terminologie

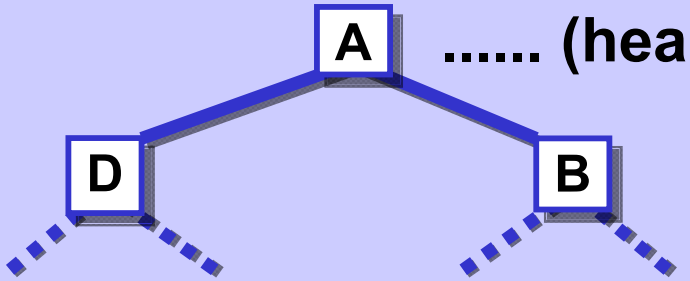


a ..... predecessor, parent of *b* *c*

..... předchůdce, rodič

b, *c* ..... successor, child of *a*

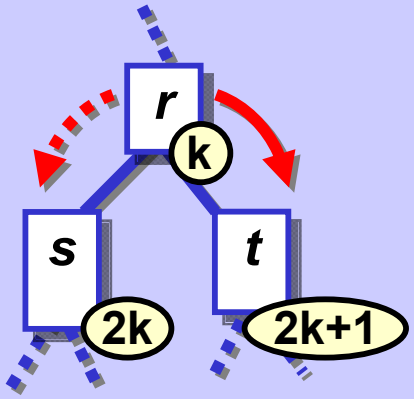
..... následník, potomek



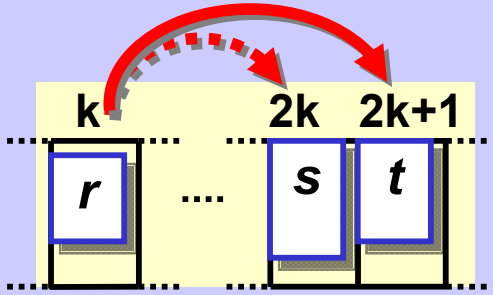
..... (heap) top ..... vrchol (haldy)

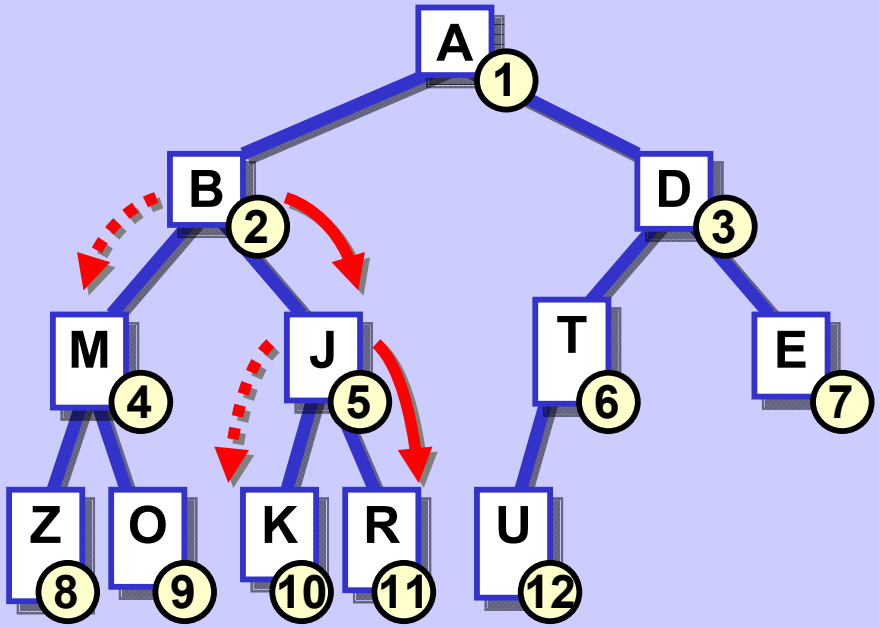
Heap sort

Halda uložená v poli

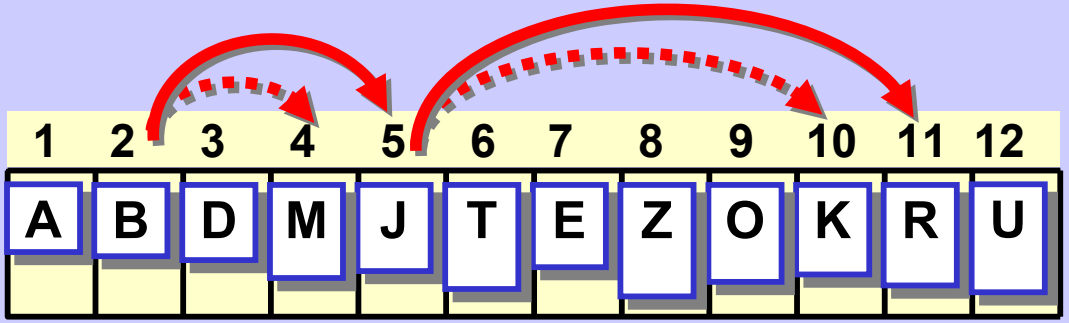


následníci





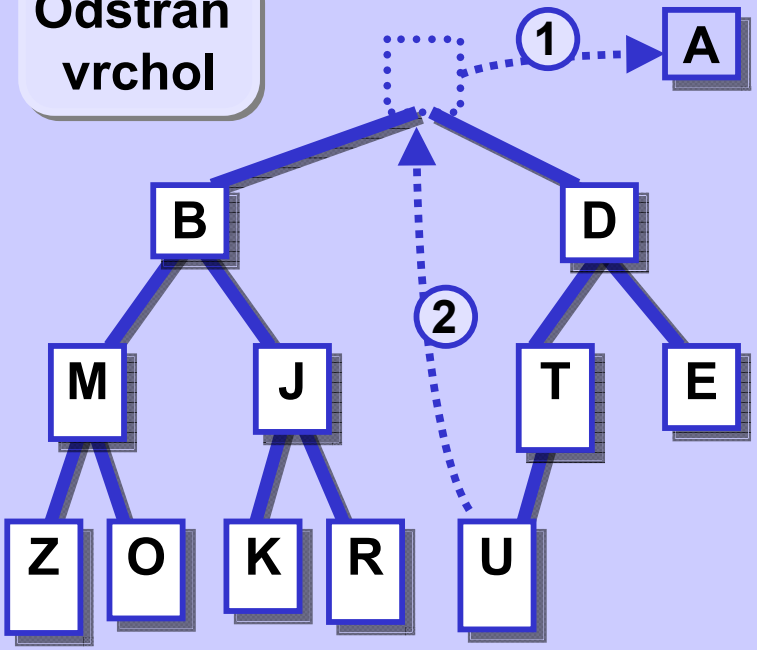
následníci



Oprava haldy

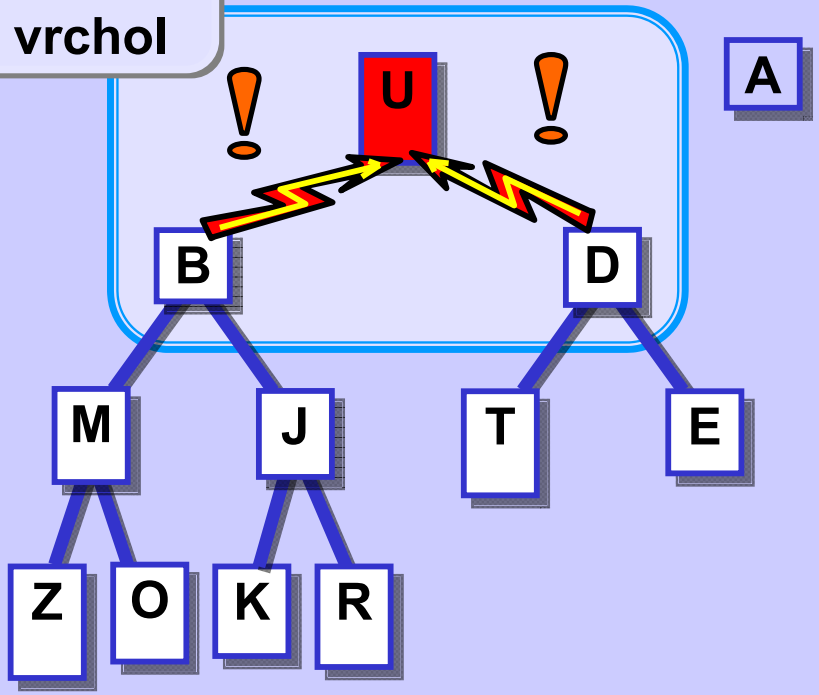
Vrchol odstraněn (1)

① Odstraň vrchol



② poslední ⟶ první

③ Vlož na vrchol



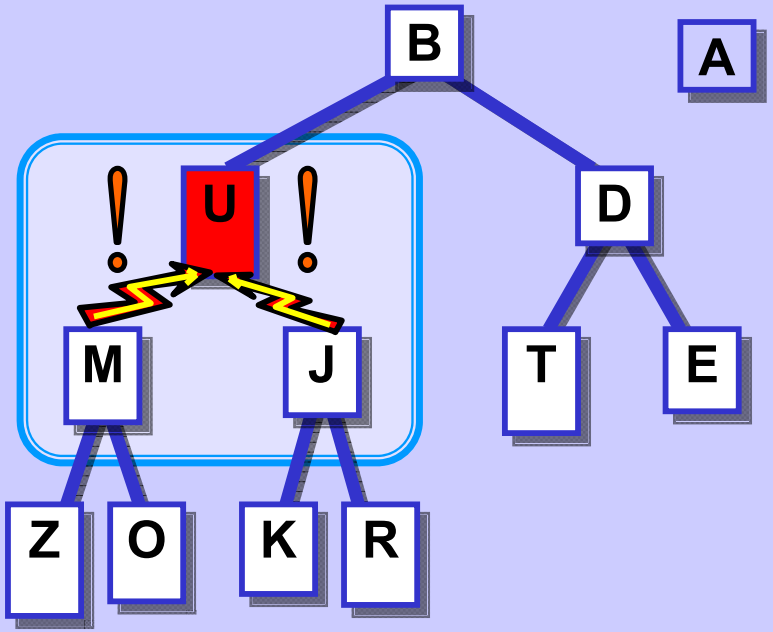
U > B, U > D, B < D

⇒ prohoď B ⟷ U

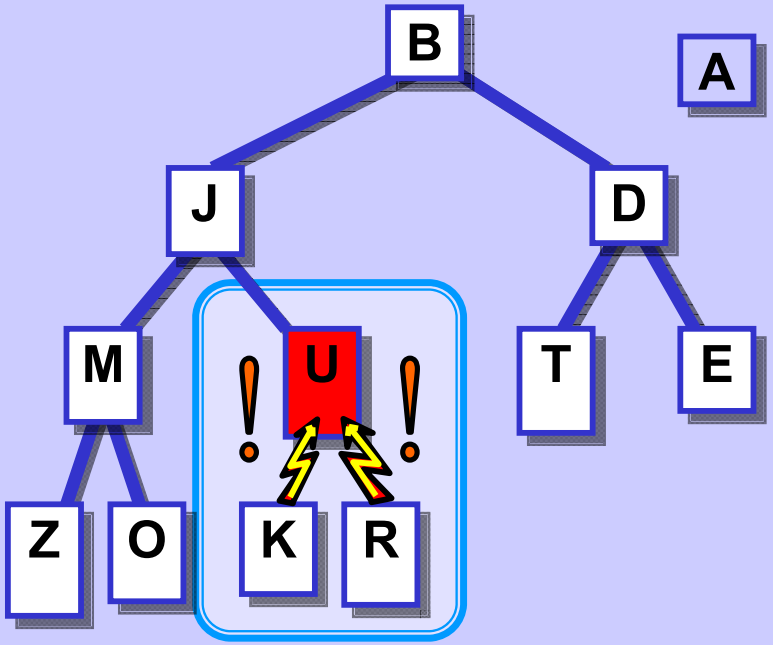
Oprava haldy

Vrchol odstraněn (2)

③ Vlož na vrchol - pokračování



U > M, U > J, J < M
⇒ prohoď J ↔ U

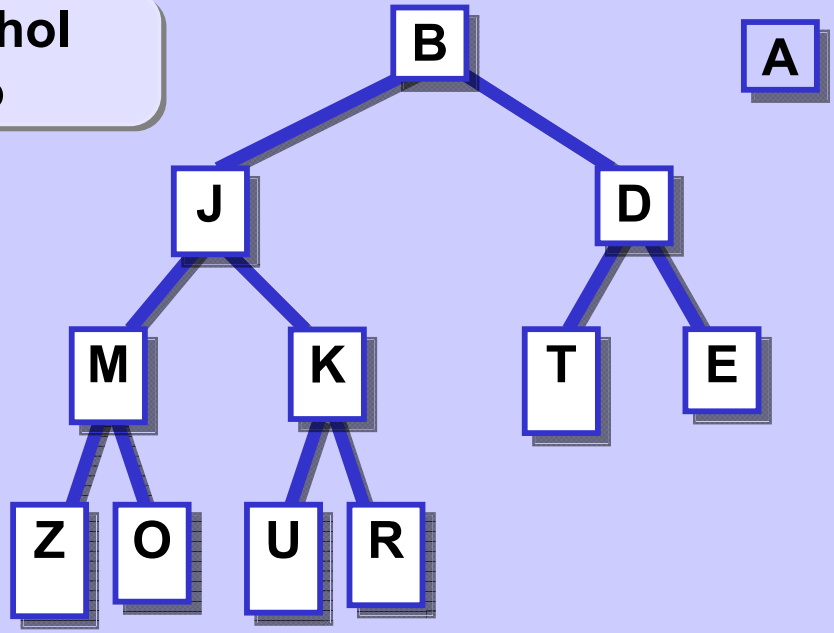


U > K, U > R, K < R
⇒ prohoď K ↔ U

Oprava haldy

Vrchol odstraněn (3)

③ Vlož na vrchol - hotovo

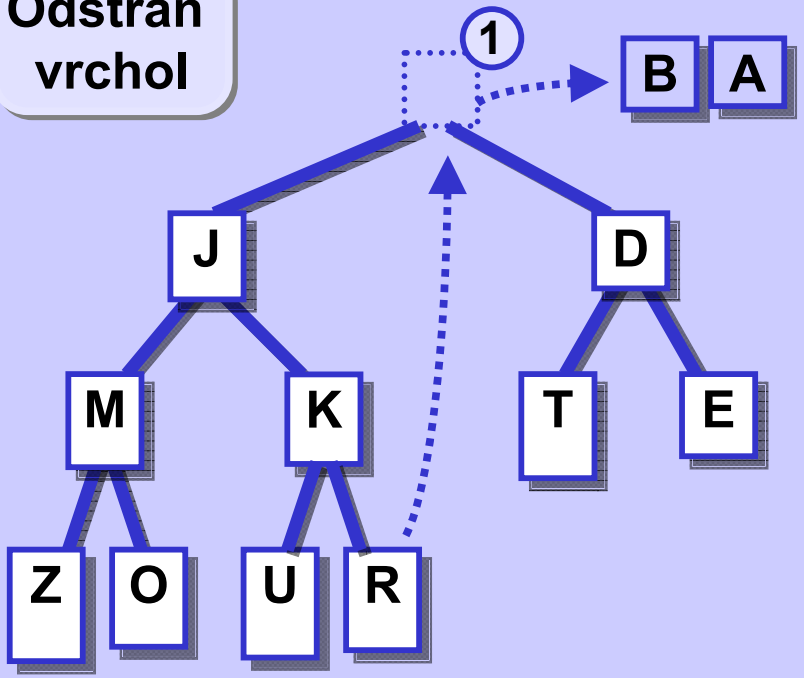


Nová halda

Oprava haldy

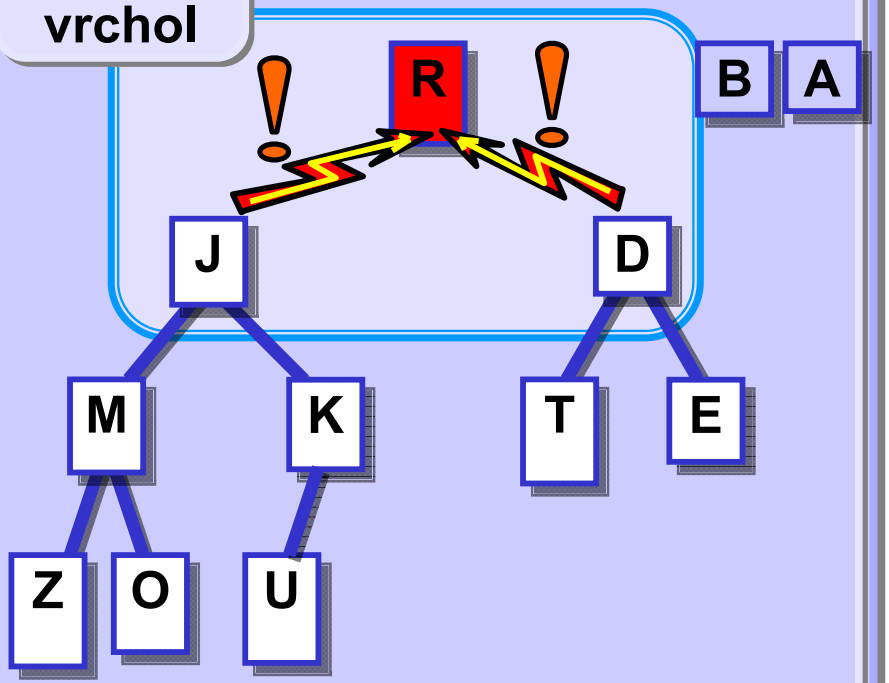
Vrchol odstraněn II (1)

① Odstraň vrchol



② poslední ⟶ první

③ Vlož na vrchol



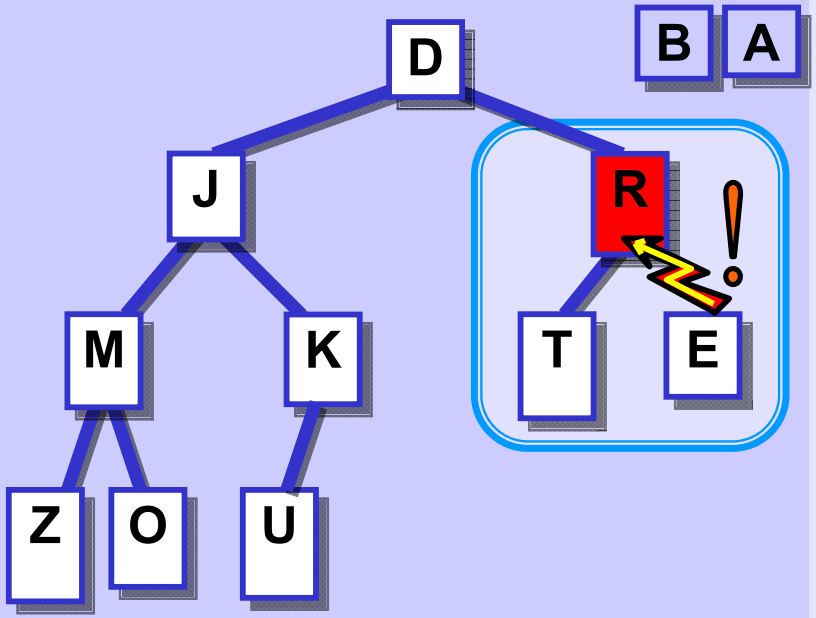
R > J, R > D, D < J

⇒ prohoď D ⟷ R

Oprava haldy

Vrchol odstraněn II (2)

③ Vlož na vrchol - pokračování

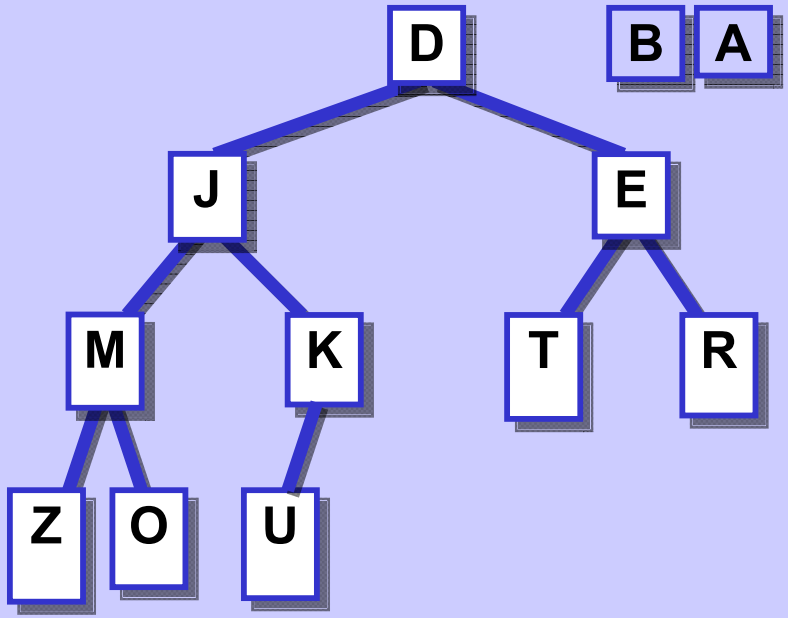


R < T, R > E

⇒ prohoď E ⟷ R

Vrchol odstraněn II (3)

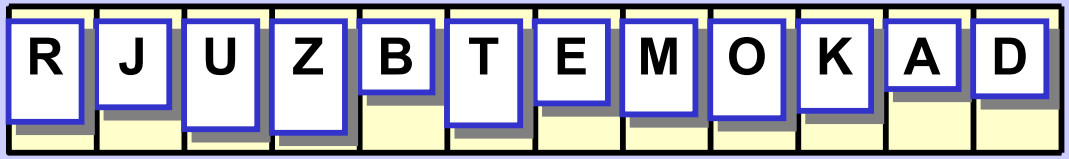
③ Vlož na vrchol - hotovo



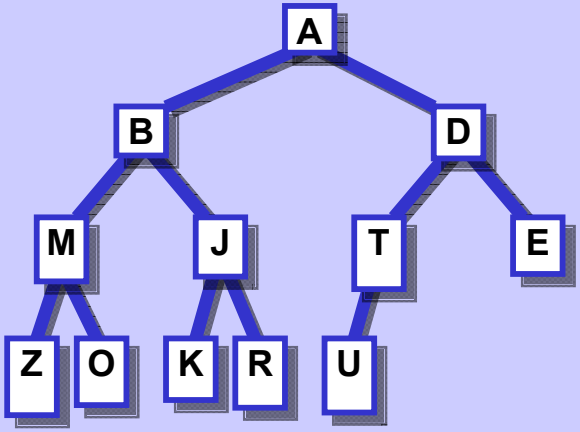
Nová halda

Heap sort

I Neseřazeno

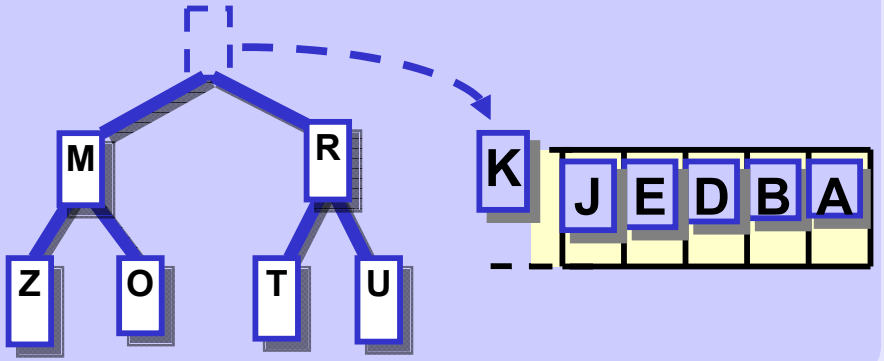


II Vytvoř haldu

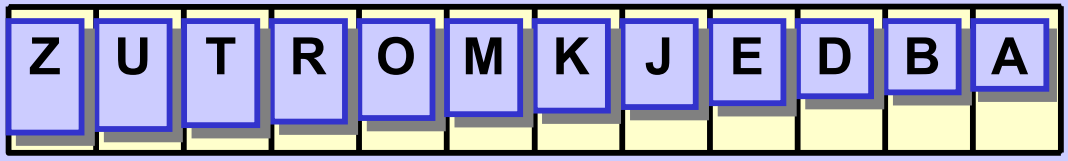


III

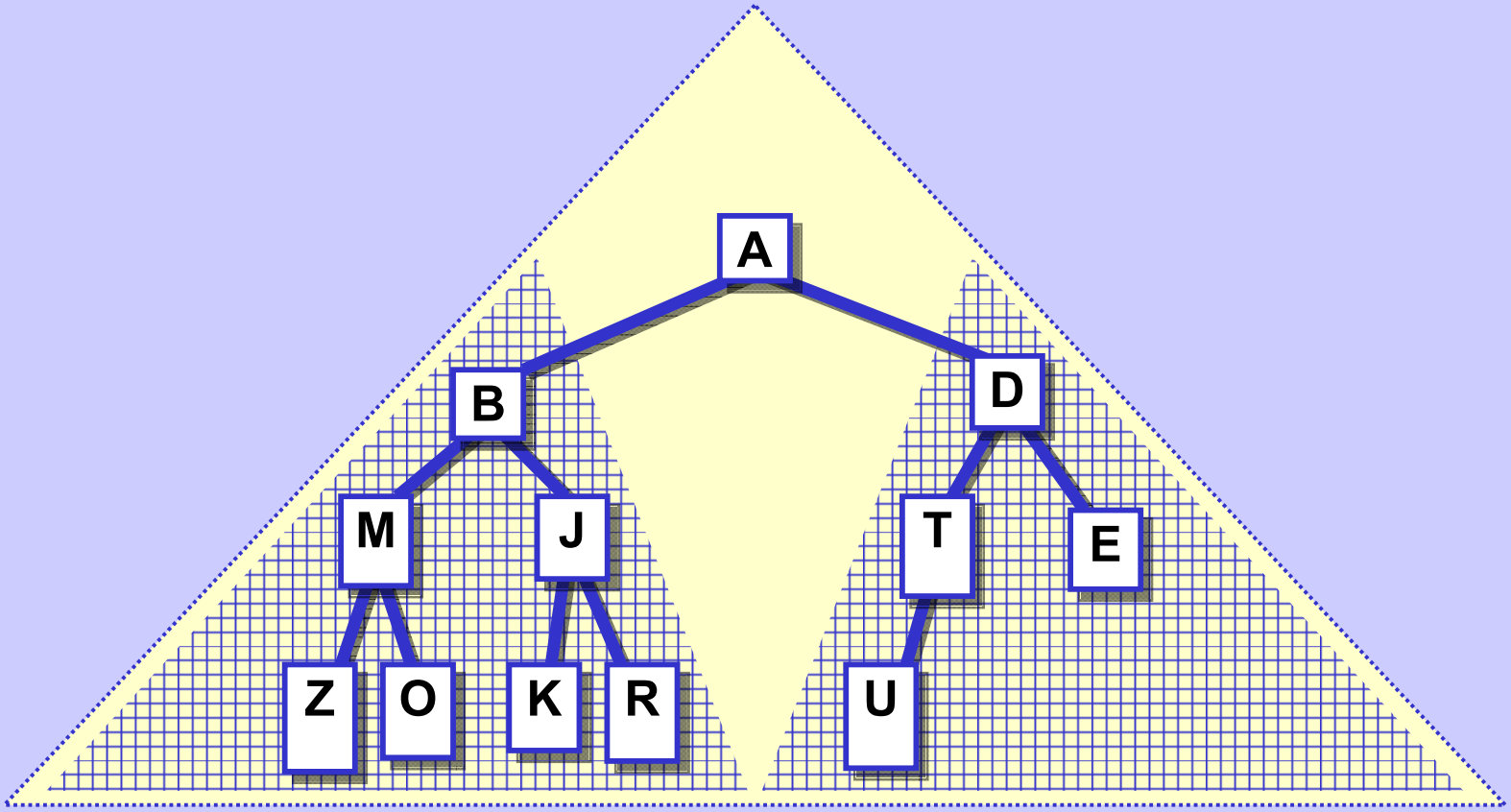
```
for (i = 0; i < n; i++)
    a[i] = “odstraň vrchol”;
```

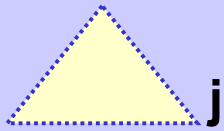


IV Seřazeno

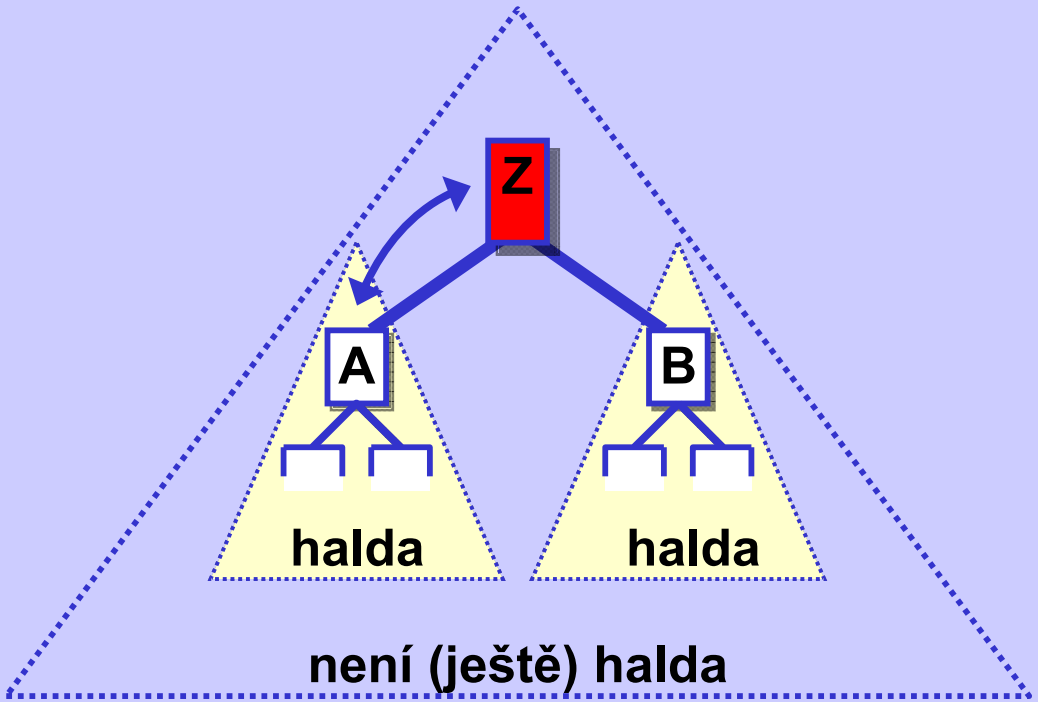


Rekurzivní vlastnost "býti haldou"



 je halda $\Rightarrow$  je halda a  je halda

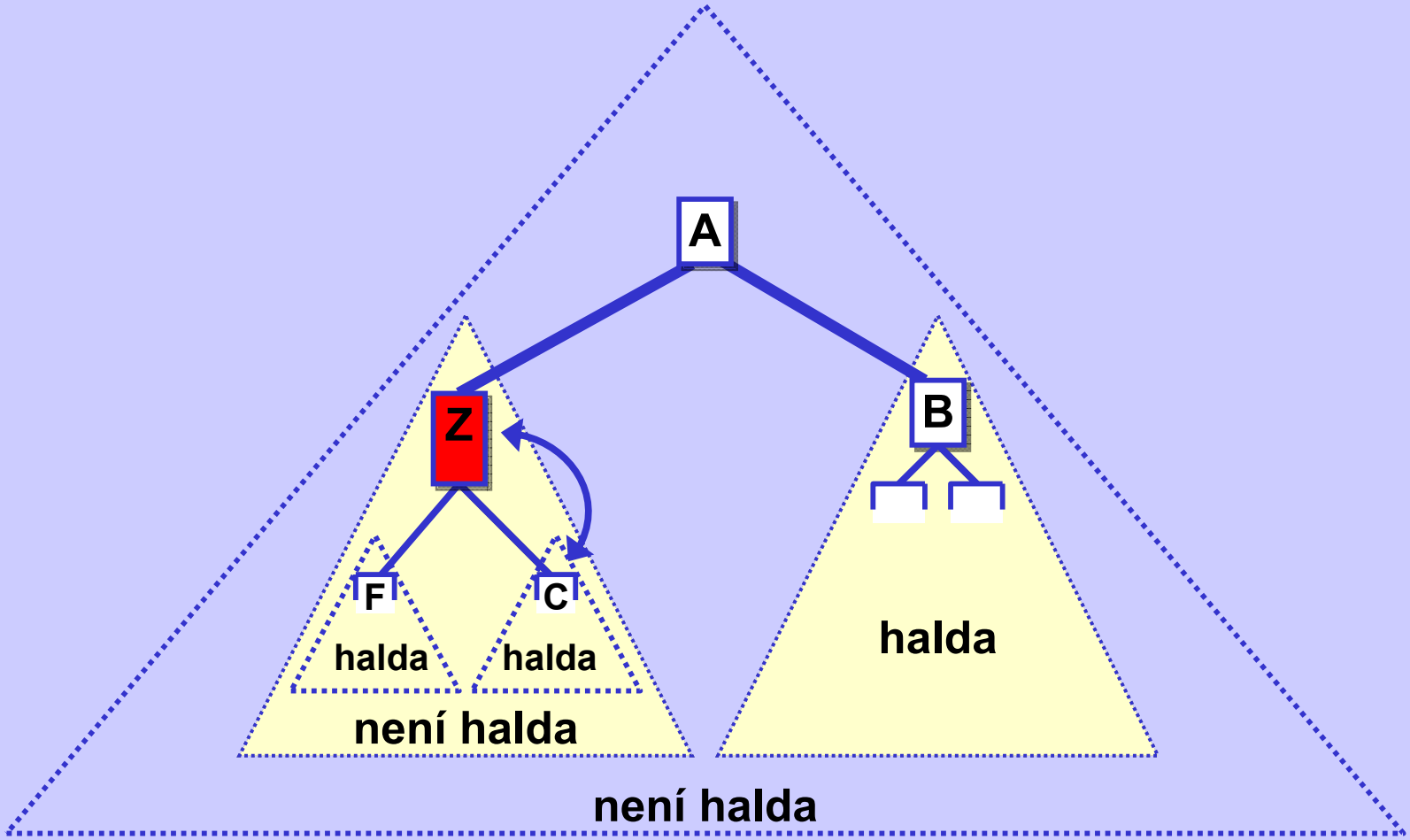
Vytvoř jednu haldu ze dvou menších



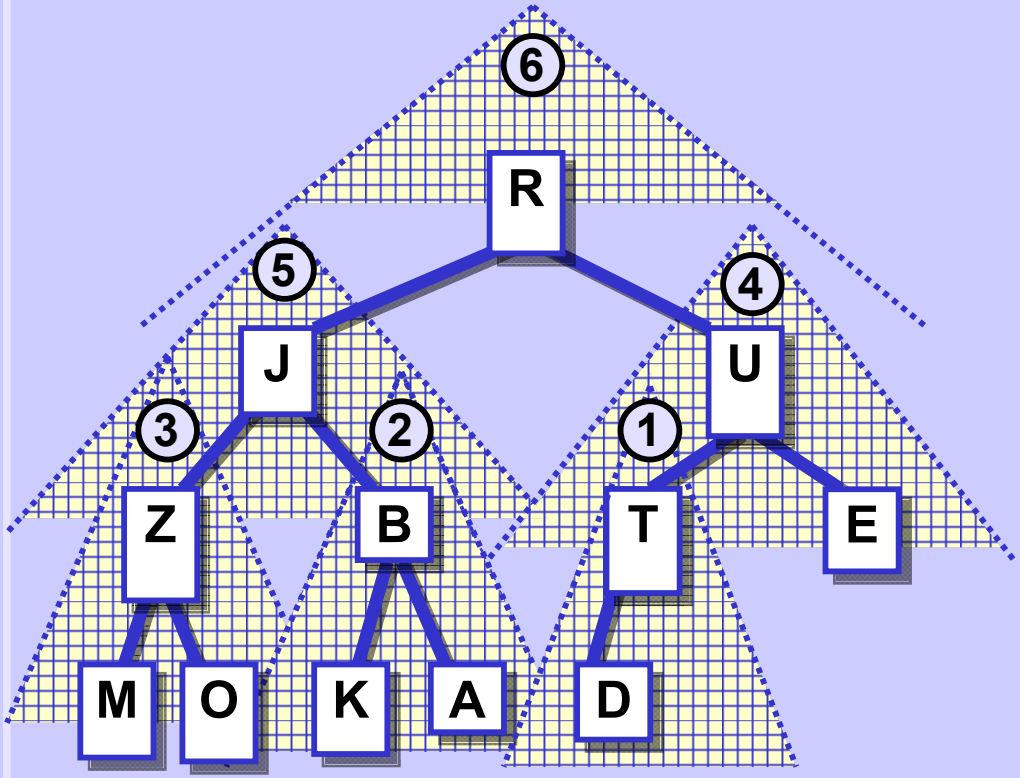
$Z > A$ nebo $Z > B$

$\Rightarrow$ prohoď: $Z \leftrightarrow \min(A,B)$

Vytvoř jednu haldu ze dvou menších



Vytvoř haldu



Vytvoř haldu v ① ...

... a vytvoř haldu v ② ...

... a vytvoř haldu v ③ ...

... a vytvoř haldu v ④ ...

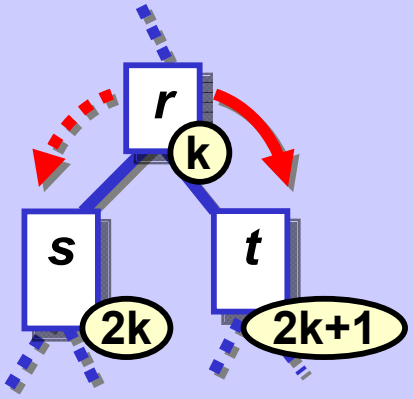
... a vytvoř haldu v ⑤ ...

... a vytvoř haldu v ⑥ ...

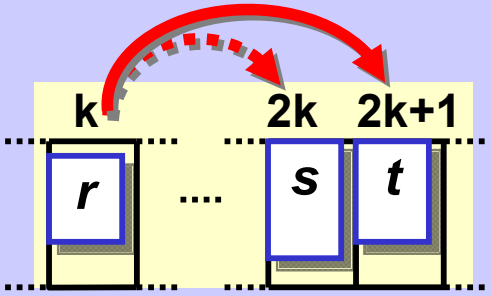
... a celá halda je hotova.

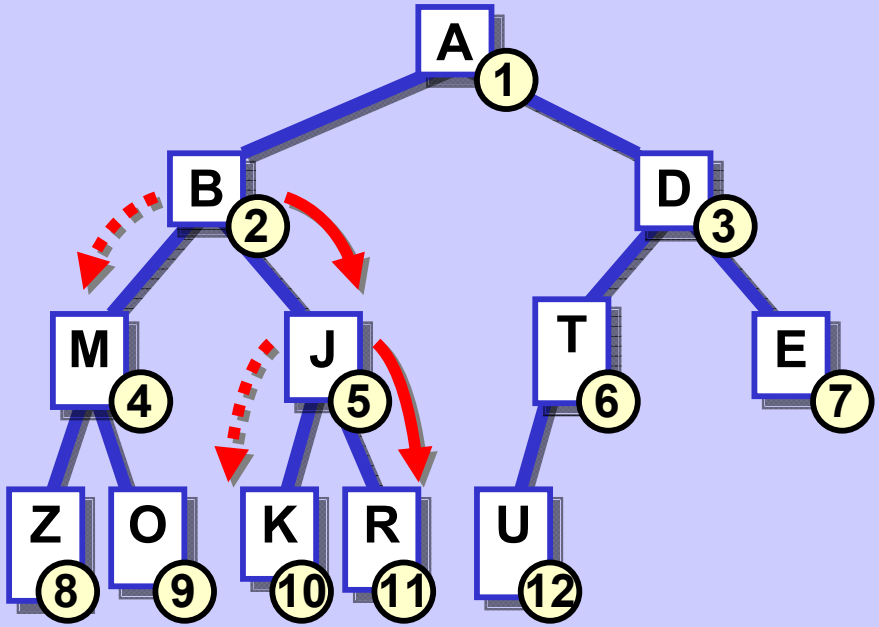
Halda v poli

Halda uložená v poli

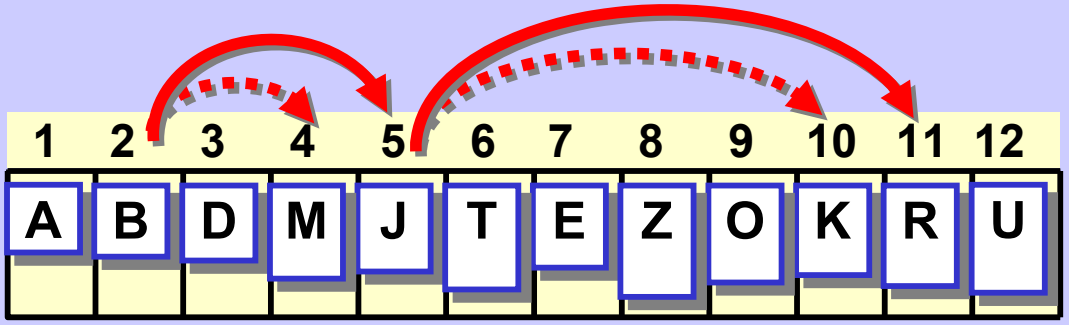


následníci



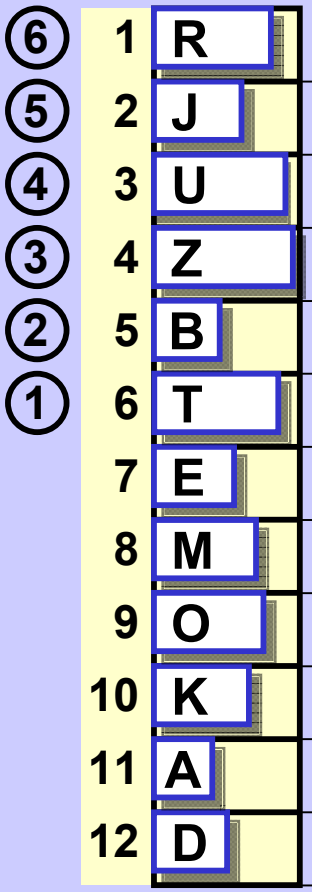


následníci



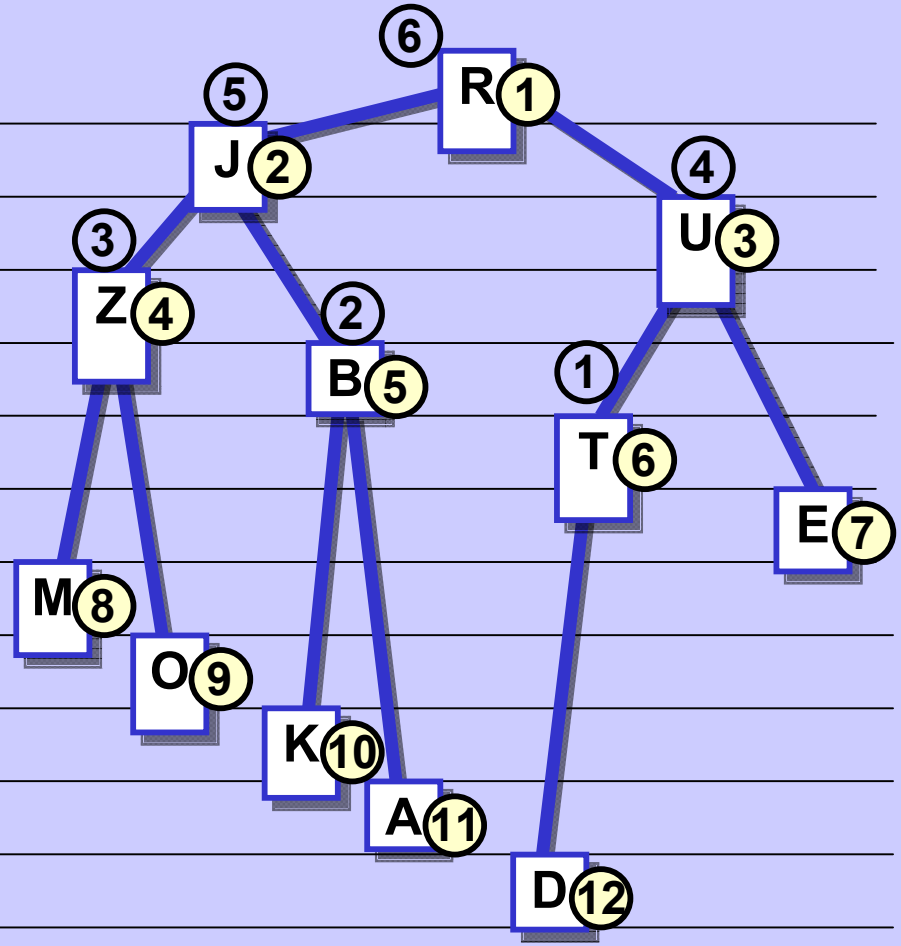
Tvorba haldy v poli

Pole



Prvky náhodně uspořádány

Není halda

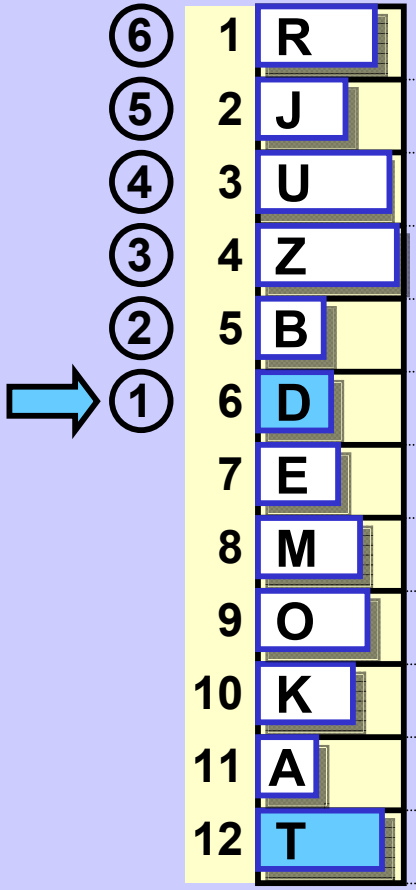


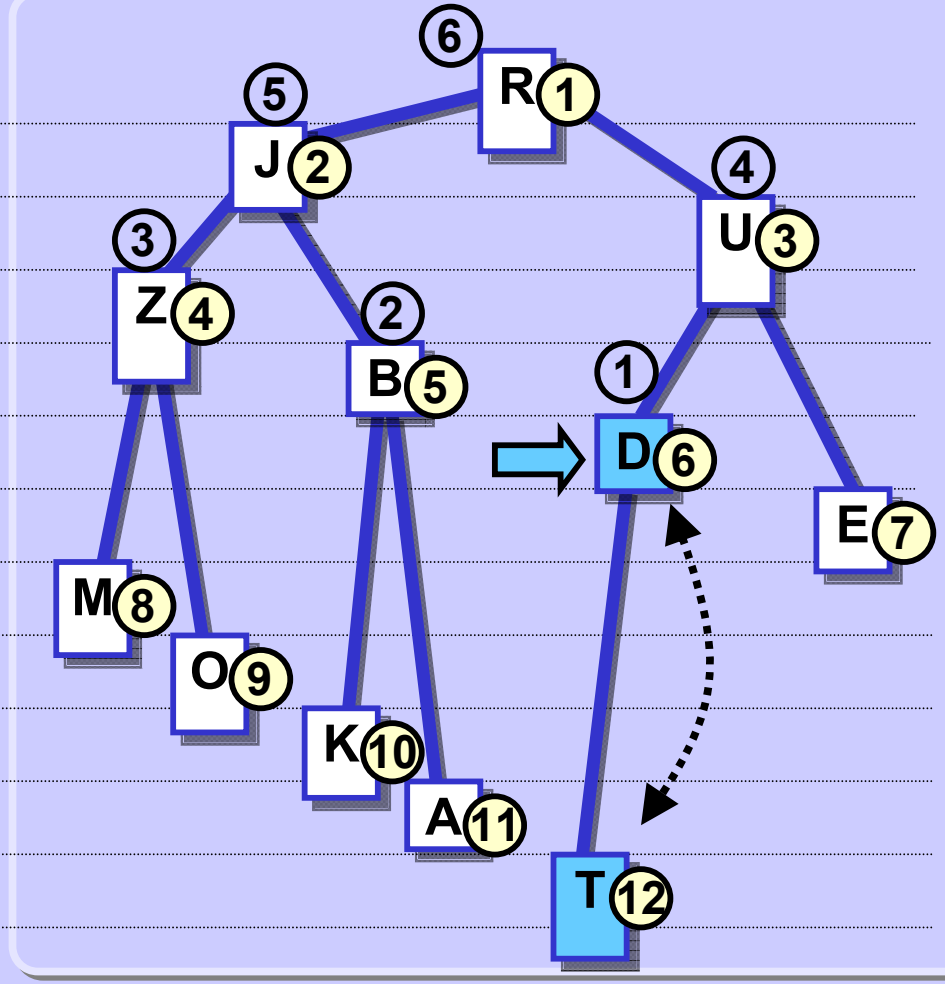
Tvorba haldy v poli

Pole

Přesuny

Aktuálně vytvořená halda





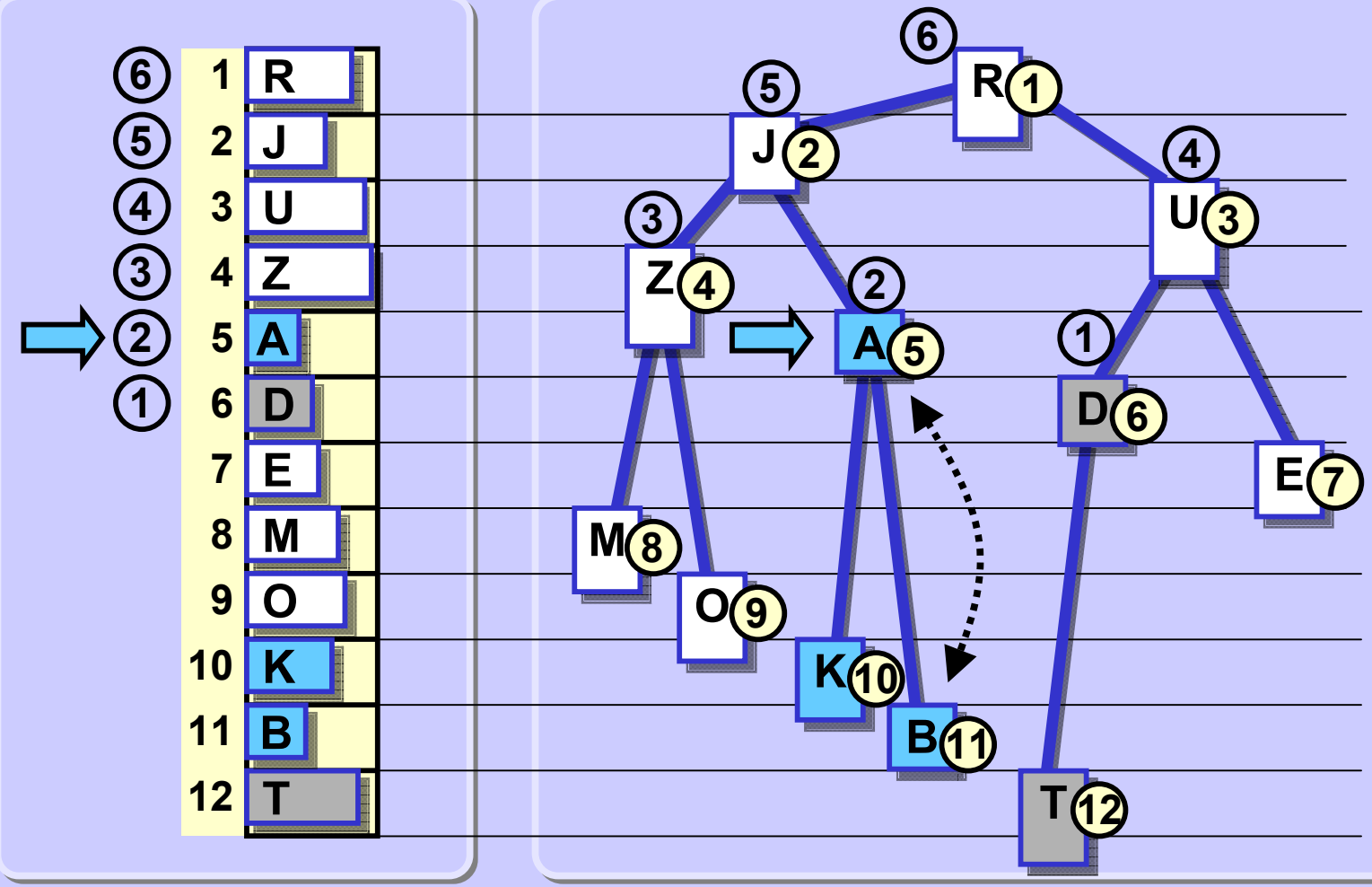
Tvorba haldy v poli

Pole

Dříve vytvořená halda

Přesuny

Aktuálně vytvořená halda



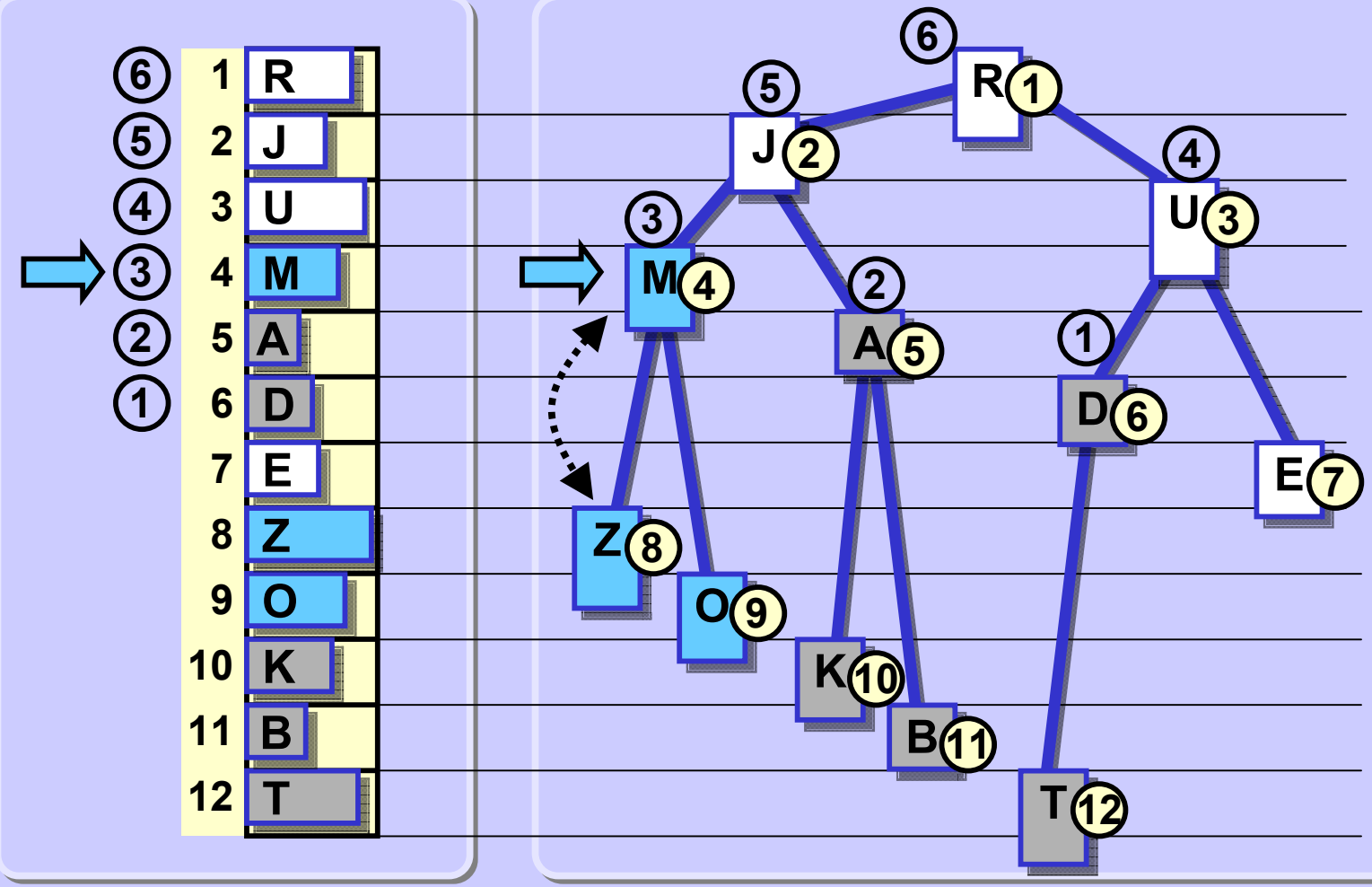
Tvorba haldy v poli

Pole

Dříve vytvořená halda

Přesuny

Aktuálně vytvořená halda



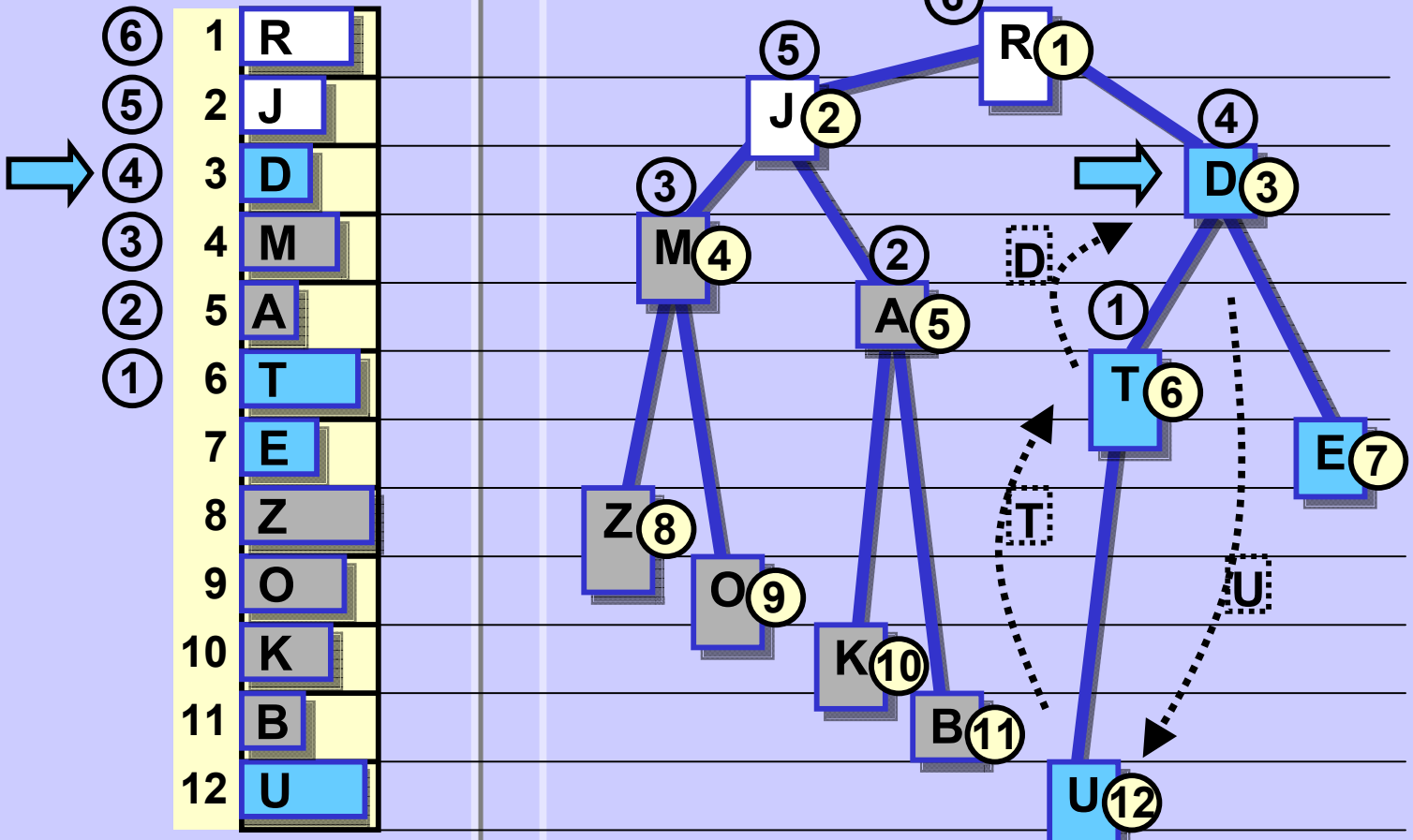
Tvorba haldy v poli

Pole

Dříve vytvořená halda

Přesuny

Aktuálně vytvořená halda

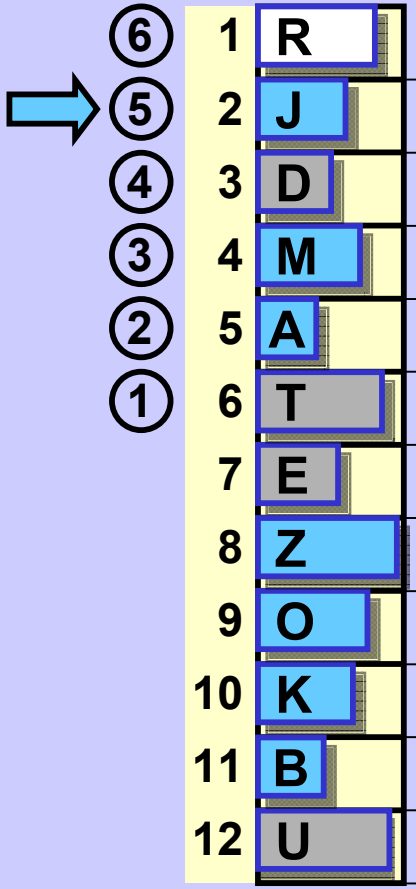


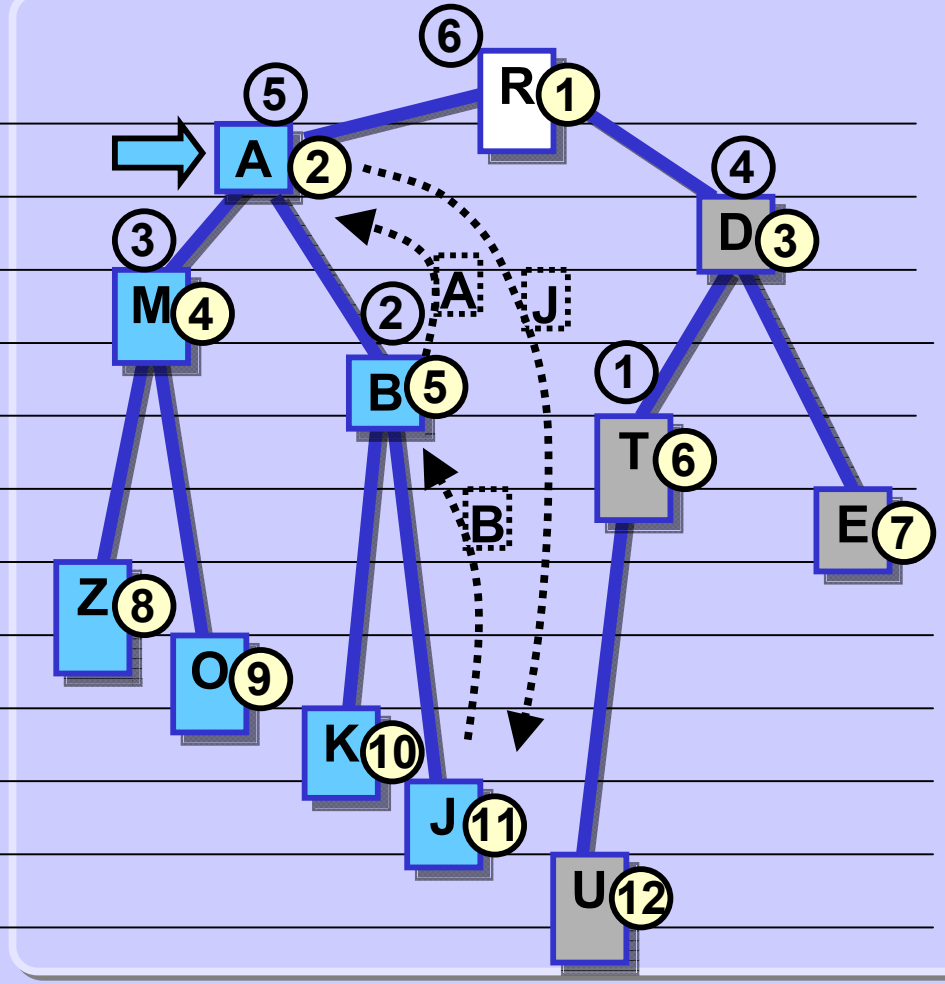
Tvorba haldy v poli

Dříve vytvořená halda ....▾ Přesuny

Aktuálně vytvořená halda

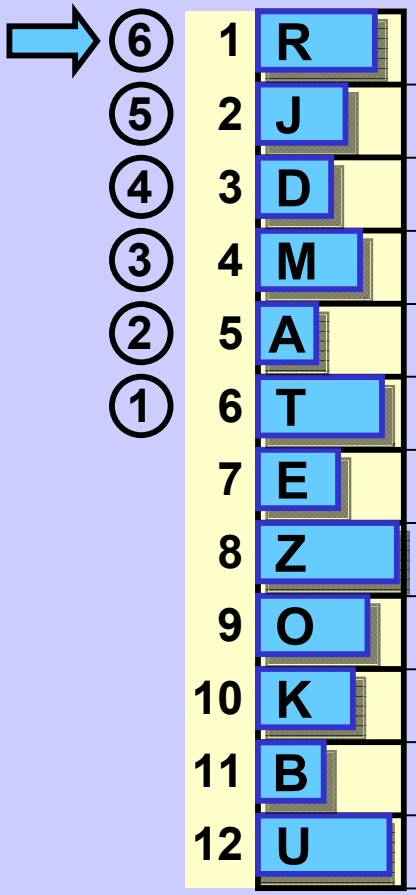
Pole





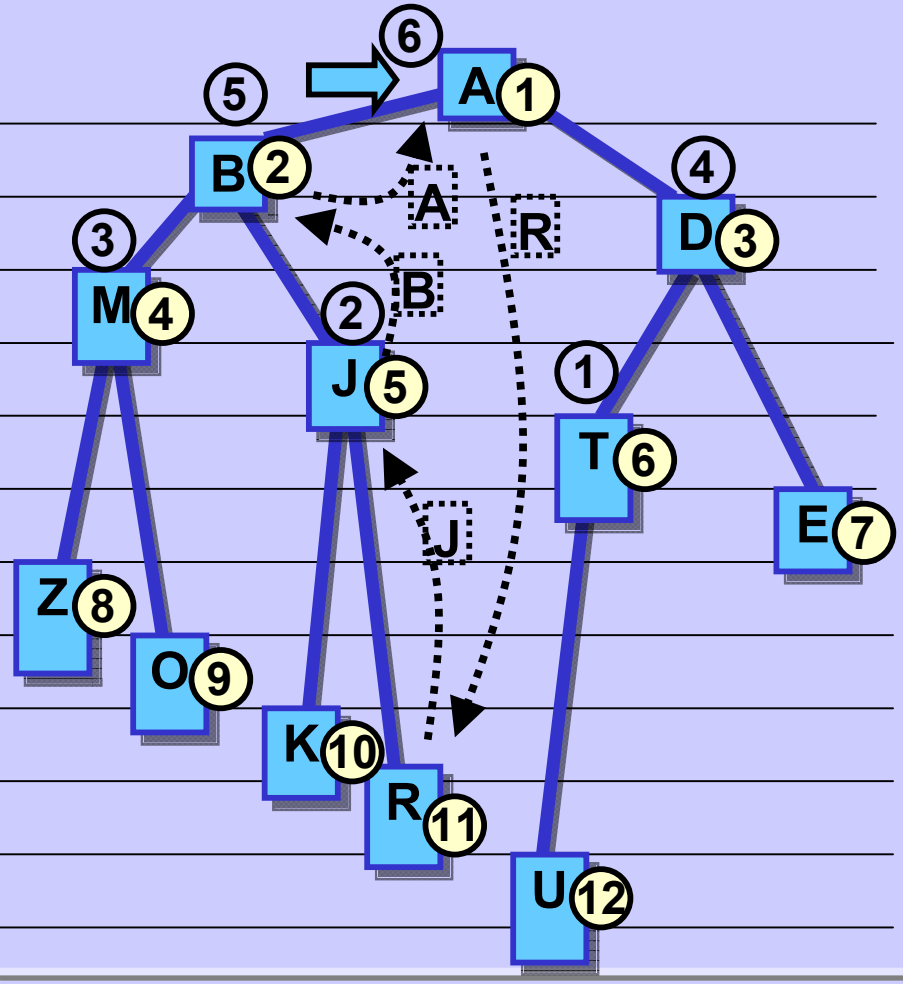
Tvorba haldy v poli

Pole



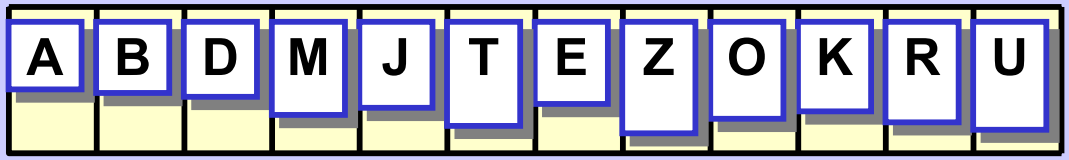
Přesuny

Aktuálně vytvořená halda

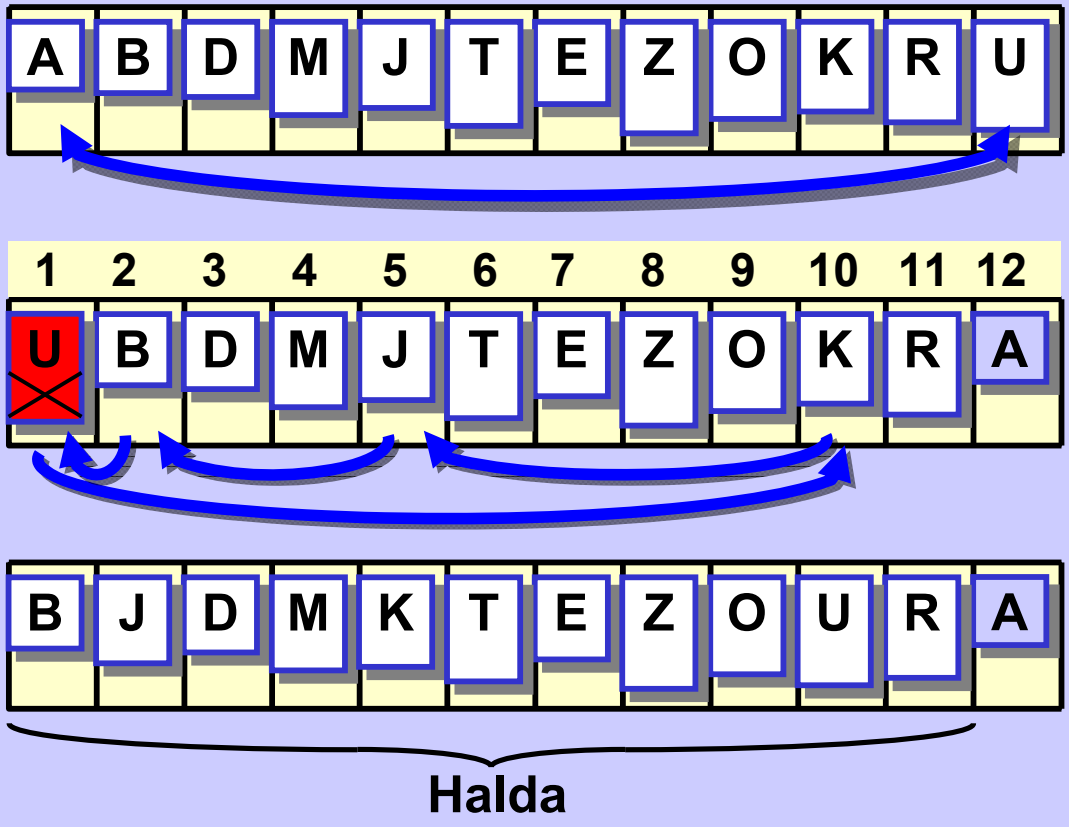


Heap sort

Halda

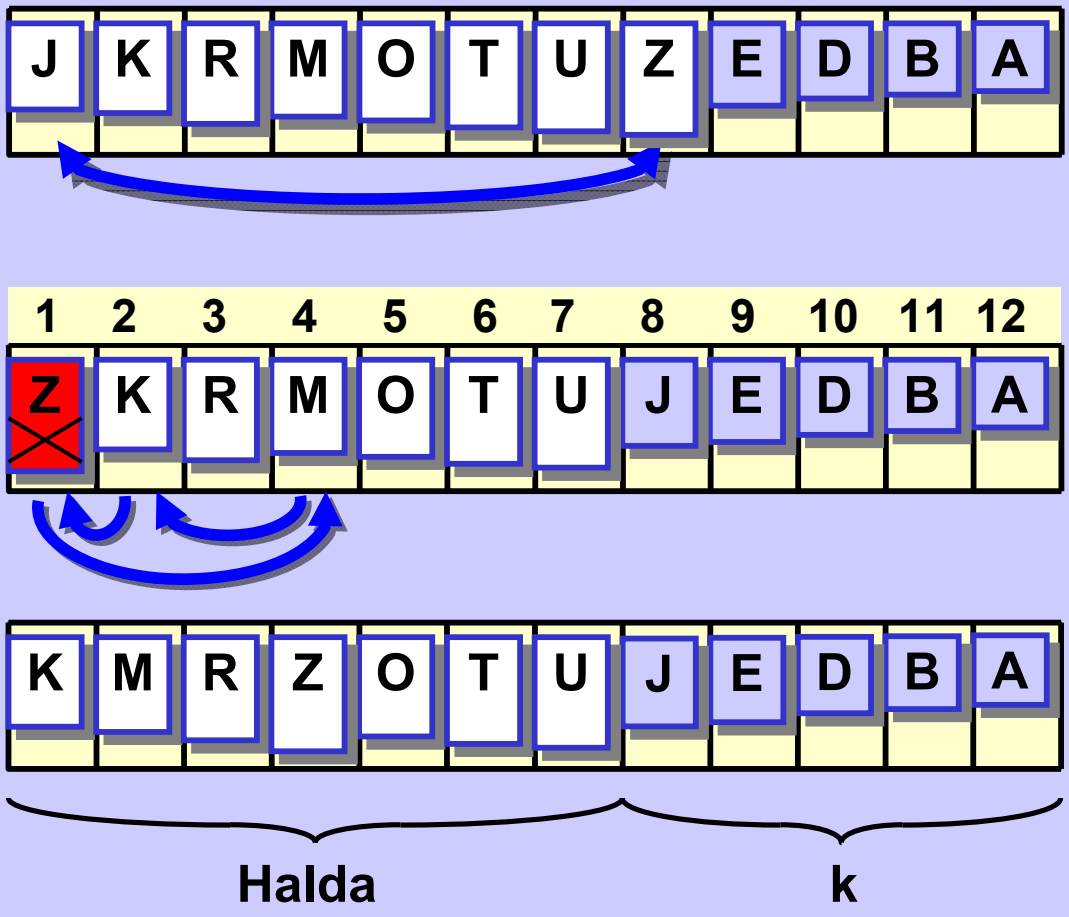


Krok 1



Heap sort

Krok k



Heap sort

```
                              // array: a[1]...a[n]  !!!!

void heapSort(Item a[], int n) {
  int i, j;
                              // create a heap
  for (i = n/2; i > 0; i--)
    repairTop(a, i, n);

                              // sort
  for (i = n; i > 1; i--) {
    swap(a, 1, i);
    repairTop(a, 1, i-1);
  }
}
```

Heap sort

```
                          // array:  a[1]...a[n]  !!!!!!
void repairTop(Item a[], int top, int bottom) {
  int i = top;            // a[2*i] and a[2*i+1]
  int j = i*2;            // are successors of a[i]

  Item topVal = a[top];

                         // try to find a successor < topVal
  if ((j < bottom) && (a[j] > a[j+1]))  j++;

                        // while (successors < topVal)
                        //         move successors up
  while ((j <= bottom) && (topVal > a[j])) {
    a[i] = a[j];
    i = j;  j = j*2; // skip to next successor
    if ((j < bottom) && (a[j] > a[j+1])) j++;
  }
  a[i] = topVal;     // put the topVal
}
```

Heap sort

repairTop operace nejhorší případ ... $\log_2(n)$ (n=velikost haldy)

vytvoř haldu ... n/2 repairTop operací

$\log_2(n/2) + \log_2(n/2+1) + ... + \log_2(n) \leq (n/2)(\log_2(n)) = \underline{O(n \cdot \log_2(n))}$

seřaď haldy ... n-1 repairTop operací, nejhorší případ:

$\log_2(n) + \log_2(n-1) + ... + 1 \leq n \cdot \log_2(n) = O(n \cdot \log_2(n))$

ale i nejlepší případ = $\underline{\Theta(n \cdot \log_2(n))}$

celkem ... vytvoř haldu + seřaď haldu = $\underline{\Theta(n \cdot \log_2(n))}$

Asymptotická složitost Heap sortu je $\Theta(n \cdot \log_2(n))$

Heap sort není stabilní