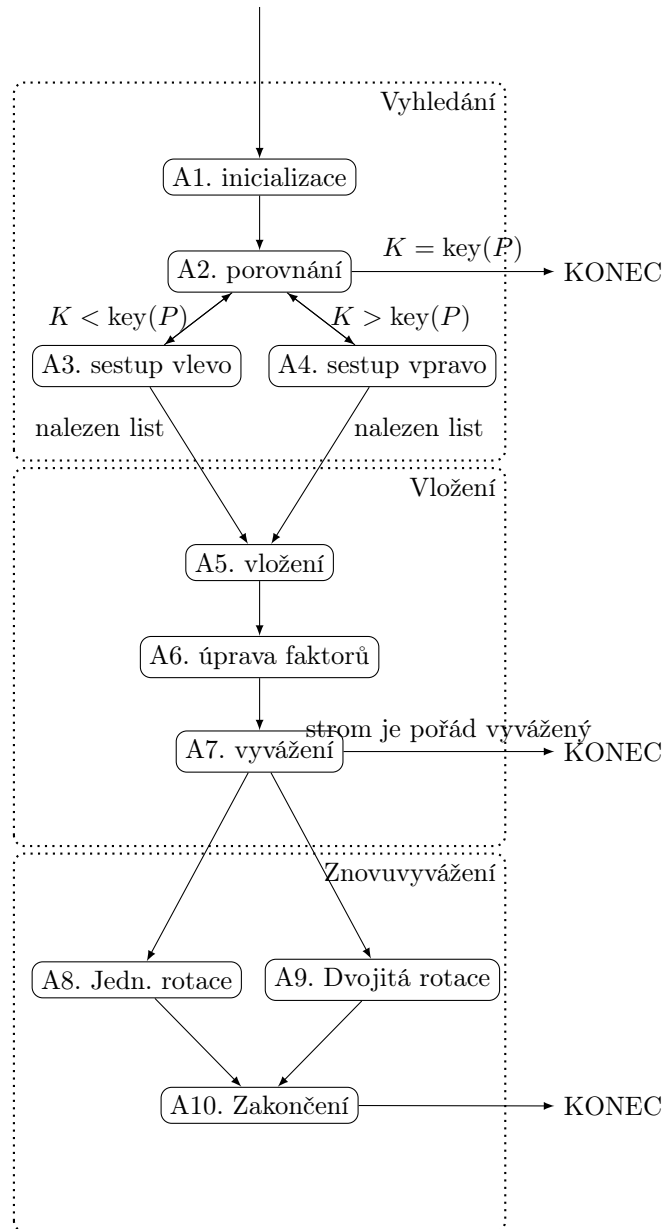


1 Vkládání do AVL-stromu



Algoritmus

P se bude pohybovat ve stromě směrem dolů.
 S bude ukazovat, kde by mohlo být nutné vyvažování.
 T bude vždycky rodič S .
 $head$ je „hlavička“,
 $right(head)$ ukazuje na kořen.

Vstup: neprázdný AVL-strom, klíč K

A1 [inicializace]

$T \leftarrow head$

$S \leftarrow right(head)$

$P \leftarrow right(head)$

A2 [porovnání]

```
    if  $K < \text{key}(P)$  then
    |   continue A3
    else if  $K > \text{key}(P)$  then
    |   continue A4
    else if  $K = \text{key}(P)$  then
    |   terminate
    end
```

A3 [pohyb vlevo]

```
     $Q \leftarrow \text{left}(P)$ .
    if  $Q = \text{NIL}$  then
    |    $Q \leftarrow \text{alloc}$ 
    |    $\text{left}(P) \leftarrow Q$ 
    |   continue A5
    else
    |   if  $B(Q) \neq 0$  then
    |   |    $T \leftarrow P$ 
    |   |    $S \leftarrow Q$ 
    |   |   end
    |   |    $P \leftarrow Q$ 
    |   |   continue A2
    end
```

A4 [pohyb vpravo]

```
     $Q \leftarrow \text{right}(P)$ 
    if  $Q = \text{NIL}$  then
    |    $Q \leftarrow \text{alloc}$ 
    |    $\text{right}(P) \leftarrow Q$ 
    |   continue A5
    else
    |   if  $B(Q) \neq 0$  then
    |   |    $T \leftarrow P$ 
    |   |    $S \leftarrow Q$ 
    |   |   end
    |   |    $P \leftarrow Q$ 
    |   |   continue A2
    end
```

A5 [vlození]

```
     $\text{key}(Q) \leftarrow K$ 
     $\text{left}(Q) \leftarrow \text{NIL}$ 
     $\text{right}(Q) \leftarrow \text{NIL}$ 
     $B(Q) \leftarrow 0$ 
```

A6 [úprava vyvažovacích faktorů] *vyvažovací faktory na uzlech mezi S a Q musí být změněny z 0 na ± 1 .*

```
    if  $K < \text{key}(S)$  then
    |    $a \leftarrow -1$ 
    else
    |    $a \leftarrow 1$ 
    end
     $R \leftarrow P \leftarrow \text{link}(a, S)$ 
    while  $P \neq Q$  do
    |   if  $K < \text{key}(P)$  then
    |   |    $B(P) \leftarrow -1$ 
    |   |    $P \leftarrow \text{left}(P)$ 
    |   else if  $K > \text{key}(P)$  then
    |   |    $B(P) \leftarrow +1$ 
    |   |    $P \leftarrow \text{right}(P)$ 
    |   end
    end
```

(link(a, S) představuje levý nebo pravý podstrom, podle toho, jestli $a = -1$ nebo $a = +1$.)

A7 [vyvažování]

```

if B( $S$ ) = 0 then
    | B( $S$ )  $\leftarrow a$ 
    | left(head)  $\leftarrow$  left(head) + 1
    | terminate
end
if B( $S$ ) =  $-a$  then B( $S$ )  $\leftarrow$  0
    terminate

if B( $S$ ) =  $a$  then
    | if B( $R$ ) =  $a$  then continue A8
    | if B( $R$ ) =  $-a$  then continue A9
end

```

A8 [jednoduchá rotace]

```

 $P \leftarrow R$ 
link( $a, S$ )  $\leftarrow$  link( $-a, R$ )
link( $-a, R$ )  $\leftarrow S$ 
B( $S$ )  $\leftarrow$  B( $R$ )  $\leftarrow$  0
continue A10

```

A9 [dvojitá rotace]

```

 $P \leftarrow$  link( $-a, R$ )
link( $-a, R$ )  $\leftarrow$  link( $a, P$ )
link( $a, P$ )  $\leftarrow R$ 
link( $a, S$ )  $\leftarrow$  link( $-a, P$ )
link( $-a, P$ )  $\leftarrow S$ 

(B( $S$ ), B( $R$ ))  $\leftarrow \begin{cases} (-a, 0), & \text{if } B(P) = a \\ (0, 0), & \text{if } B(P) = 0 \\ (0, a), & \text{if } B(P) = -a \end{cases}$ 

B( $P$ )  $\leftarrow$  0
continue A10.

```

A10 [zakončení]

```

if  $S = \text{right}(T)$  then
    | right( $T$ )  $\leftarrow P$ 
else
    | left( $T$ )  $\leftarrow P$ 
end

```

Abychom mohli studovat chování vkládání a vyvažovacích fází algoritmu, klasifikujeme externí uzly vyvážených algoritmů následovně:

Cesta vedoucí z externího uzlu k prvnímu uzlu, který má vyvažovací faktor 0, může být specifikována sekvencemi $+$ a $-$:

- $+$ pro pravý podstrom
- $-$ pro levý podstrom.

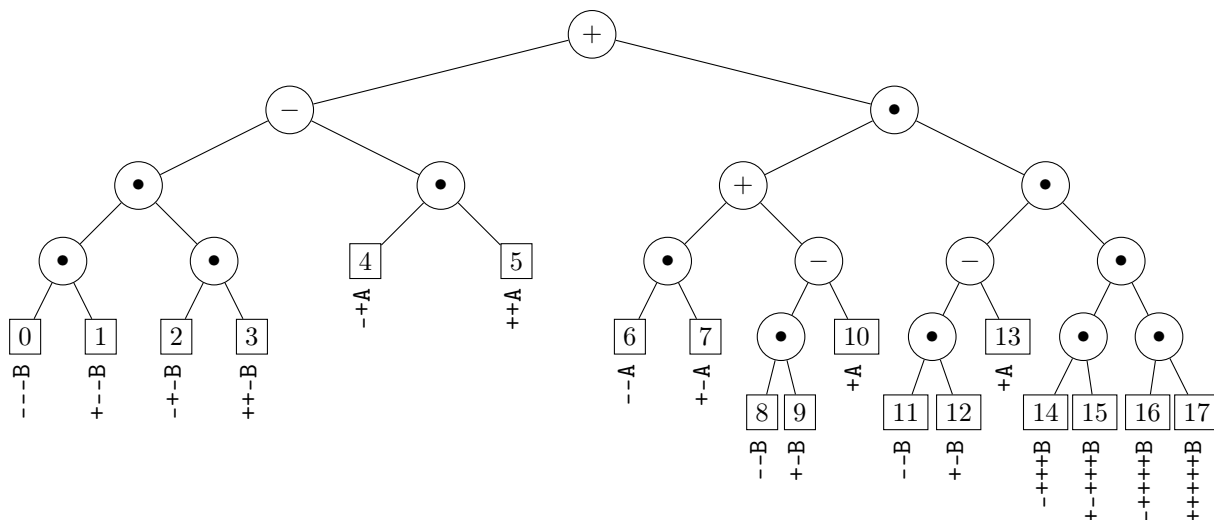
Píšeme tuto sekvenci dokud nenarazíme na uzlu s nenulovým vyvažovacím faktorem, nebo na kořen.

Poté zapíšeme A nebo B podle toho, jestli strom vzniklý umístěním nového vnitřního uzlu na místo onoho externího bude vyvážený nebo ne.

Takže:

- Specifikace končící A nepotřebuje rotaci.
- Specifikace končící $--B$, $++B$ potřebuje jednoduchou rotaci.
- Specifikace končící $+-B$, $-+B$ potřebuje jednoduchou rotaci.
- Specifikace obsahuje k znamének, je potřeba upravit $k - 1$ vyvažovacích faktorů.

Tyto specifikace ukazují základní fakta, která určují čas potřebný pro kroky A6–A10.



Empirické testy, na náhodných číslech $100 \leq n \leq 2000$ dávají odhady pravděpodobností uvedené v tabulce:

délka cesty k	bez vyvažování	jednoduchá rotace	dvojitá rotace
1	0.143	0.000	0.000
2	0.152	0.143	0.143
3	0.092	0.048	0.048
4	0.060	0.024	0.024
5	0.036	0.010	0.010
> 5	0.051	0.009	0.008

Nepřesný model

Mějme následující (nepřesný) model pro algoritmus AVL.

Předpokládejme, že p je pravděpodobnost, že vyvažovací faktor daného uzlu ve velkém stromu vybudovaném algoritmem A je 0.

Pak pravděpodobnost že tento faktor je 1 je $\frac{1}{2}(1-p)$ a pravděpodobnost že tento faktor je -1 je $\frac{1}{2}(1-p)$.

Dále uvažujme (bez odůvodnění), že vyvažovací faktory všech uzlů jsou nezávislé.

Pak pravděpodobnost, že krok A6 nastaví přesně $k-1$ vyvažovacích faktorů na nenulovou hodnotu je $p^{k-1}(1-p)$ takže průměrná hodnota k je $1/(1-p)$.

Pravděpodobnost, že musíme zrotovat část stromu je $q \approx \frac{1}{2}$.

Vložení nového uzlu by mělo zvýšit počet vyvážených uzlů o p – toto číslo je zvýšeno

- o 1 v kroku A5,
- o $-p/(1-p)$ v kroku A6
- o q v kroku A7
- o $2q$ v kroku A8 nebo A9

takže bychom měli mít

$$p = 1 - p/(1-p) + 3q \approx 5/2 - p/(1-p).$$

Řešení p dává

$$p \approx \frac{9 - \sqrt{41}}{4} \quad 1/(1-p) \approx 2.851.$$

(víceméně ve shodě s tou tabulkou s pravděpodobnostmi)

2 Reprezentace lineárního seznamu

2.1 Algoritmus vyhledávání dle pozice

Vstup: neprázdný AVL-strom, klíč K

B1 [inicializace]

$M \leftarrow k$

$P \leftarrow \text{right}(\text{head})$

```

B2 [porovnání]
    if  $P = NIL$  then FAIL
    else if  $M < \text{rank}(P)$  then continue B3
    else if  $M > \text{rank}(P)$  then continue B4
    else if  $M = \text{rank}(P)$  then SUCCESS

B3 [sestup vlevo]
     $P \leftarrow \text{left}(P)$ 
    continue B2

B4 [sestup vpravo]
     $M \leftarrow M - \text{rank}(P)$ 
     $P \leftarrow \text{right}(P)$ 
    continue B2

```

2.2 Algoritmus vkládání na pozici

Vstup: lineární seznam reprezentovaný jako AVL-strom, číslo k , ukazatel na nový uzel Q

```

C1 [inicializace]
     $T \leftarrow \text{head}$ 
     $S \leftarrow P \leftarrow \text{right}(NIL)$ 
     $U \leftarrow M \leftarrow k$ 

C2 [porovnání]
    if  $M \leq \text{rank}(P)$  then
        | continue C3
    else
        | continue C4
    end

C3 [sestup vlevo]
     $\text{rank}(P) \leftarrow \text{rank}(P) + 1$ 
     $R \leftarrow \text{left}(P)$ 
    if  $R = NIL$  then
        |  $\text{left}(P) \leftarrow Q$ 
        | continue C5
    else
        if  $B(R) \neq 0$  then
            |  $T \leftarrow P$ 
            |  $S \leftarrow R$ 
            |  $U \leftarrow M$ 
        end
         $P \leftarrow R$ 
        continue C2
    end

C4 [sestup vpravo]
     $M \leftarrow M - \text{rank}(P)$ 
     $R \leftarrow \text{right}(P)$ 
    if  $R = NIL$  then
        | nastav  $\text{right}(P) \leftarrow Q$  jdi na C5.
    else
        if  $B(R) \neq 0$  then
            |  $T \leftarrow P$ 
            |  $S \leftarrow R$ 
            |  $U \leftarrow M$ 
        end
         $P \leftarrow R$ 
        continue C2
    end
end

```

C5 [vložení]

```
rank( $Q$ )  $\leftarrow$  1  
left( $Q$ )  $\leftarrow$  right( $Q$ )  $\leftarrow$  NIL  
 $B(Q) \leftarrow 0$ 
```

C6 [úprava vyv. faktorů]

```
 $M \leftarrow U$   
(to nastaví původní hodnotu  $M$  – tj. když  $P$  bylo  $S$ )  
if  $M < \text{rank}(S)$  then  
|  $R \leftarrow P \leftarrow \text{left}(S)$   
|  $a \leftarrow -1$   
else  
|  $R \leftarrow \text{Pright}(S)$   
|  $a \leftarrow +1$   
|  $M \leftarrow M + \text{rank}(S)$   
end  
while  $P \neq Q$  do  
| if  $M < \text{rank}(P)$  then  
| |  $B(P) \leftarrow -1$   
| |  $P \leftarrow \text{left}(P)$   
| else if  $M > \text{rank}(P)$  then  
| |  $B(P) \leftarrow +1$   
| |  $M \leftarrow M - \text{rank}(P)$   
| |  $P \leftarrow \text{left}(P)$   
| end  
| (pokud  $M = \text{rank}(P)$ , pak  $P = Q$  a pokračujeme C7)  
end
```

C7 [vyvažování]

```
if  $B(S) = 0$  then  
|  $B(S) \leftarrow a$   
| left(head)  $\leftarrow$  left(head) + 1  
| terminate  
else if  $B(S) = -a$  then  
|  $B(S) \leftarrow 0$   
| terminate  
else if  $B(S) = a$  then  
| if  $B(R) = a$  then continue C8  
| if  $B(R) = -a$  then continue C9  
end
```

C8 [jednoduchá rotace]

```
 $P \leftarrow R$   
link( $a, S$ )  $\leftarrow$  link( $-a, R$ )  
link( $-a, R$ )  $\leftarrow S$   
 $B(S) \leftarrow B(R) \leftarrow 0$   
if  $a = +1$  then rank( $R$ )  $\leftarrow$  rank( $R$ ) + rank( $S$ )  
if  $a = -1$  then rank( $S$ )  $\leftarrow$  rank( $S$ ) - rank( $R$ )  
continue C10
```

C9 [dvojitá rotace]

A9 [dvojitá rotace]

```

$$P \leftarrow \text{link}(-a, R)$$

$$\text{link}(-a, R) \leftarrow \text{link}(a, P)$$

$$\text{link}(a, P) \leftarrow R$$

$$\text{link}(a, S) \leftarrow \text{link}(-a, P)$$

$$\text{link}(-a, P) \leftarrow S$$

$$(\text{B}(S), \text{B}(R)) \leftarrow \begin{cases} (-a, 0), & \text{if } \text{B}(P) = a \\ (0, 0), & \text{if } \text{B}(P) = 0 \\ (0, a), & \text{if } \text{B}(P) = -a \end{cases}$$

$$\text{B}(P) \leftarrow 0$$
if  $a = +1$  then

- |  $\text{rank}(R) \leftarrow \text{rank}(R) - \text{rank}(P)$
- |  $\text{rank}(P) \leftarrow \text{rank}(P) + \text{rank}(S)$

endif  $a = -1$  then

- |  $\text{rank}(P) \leftarrow \text{rank}(P) + \text{rank}(R)$
- |  $\text{rank}(S) \leftarrow \text{rank}(S) - \text{rank}(P)$

end  
continue C10.
```

C10 [dokončení]

```
if  $S = \text{right}(T)$  then

- |  $\text{right}(T) \leftarrow P$

else

- |  $\text{left}(T) \leftarrow P$

end  
.
```

3 Optimální stromy

Pokud je každý klíč vyhledáván stejně často, je samozřejmě nejlepší vyvážený strom. Teď se zaměříme na případ, kdy jsou frekvence vyhledávání různé.

Mějme klíče Uvažujme pravděpodobnosti $p_1, \dots, p_n$ a pravděpodobnosti $q_0, q_1, \dots, q_n$, kde

- p_i je pravděpodobnost hledání i -tého vloženého klíče K_i ,
- q_i je pravděpodobnost hledání klíče, který leží mezi klíči K_i a K_{i+1} ,
- q_i je pravděpodobnost hledání klíče, který leží před K_1 ,
- q_n je pravděpodobnost hledání klíče, který leží před K_n ,

tak že $p_1 + p_2 + \dots + p_n + q_0 + q_1 + \dots + q_n = 1$.

Chceme najít BST (optimální strom), který minimalizuje počet porovnání během hledání:

$$\sum p_j(h(u_j) + 1) + \sum q_k(h(u_k))$$

kde j_k je j -tý interní uzel a u_k je $(k+1)$ -tý externí uzel; $h(\cdot)$ je hloubka (délka cesty od kořene).

Pozorování: Podstromy optimálního stromu jsou optimální.

Označme (pro $0 \leq j \leq n$)

- $c(i, j)$ – cena optimálního podstromu s vahami $(p_{i+1}, \dots, p_j; q_i, \dots, q_j)$.
- $w(i, j)$ – součet těchto vah, tj. $p_{i+1} + \dots + p_j + q_i + \dots + q_j$.

Z toho dostáváme, že

$$c(i, i) = 0 \tag{1}$$

$$c(i, j) = w(i, j) + \min_{i < k \leq j} (c(i, k-1) + c(k, j)) \quad \text{pro } i < j, \tag{2}$$

protože minimální možná cena stromu s kořenem u_k je

$$w(i, j) = c(i, k-1) + c(k, j).$$

Pro $i < j$, nechť $R(i, j)$ je množina všech k , pro které je v (2) dosaženo minimum (tj. množina možných kořenů optimálních stromů).

Rovnice (2) umožňuje vyhodnotit $c(i, j)$ pro $j-1 = 1, 2, \dots, n$. Je asi $\frac{1}{2}n^2$ takových hodnot. Minimalizace je prováděna pro $\frac{1}{6}n^3$ hodnot k . To znamená že můžeme najít optimální strom v čase $\mathcal{O}(n^3)$ a paměti $\mathcal{O}(n^2)$.

Ve skutečnosti jsme na tom lépe. Nepotřebujeme počítat celou množinu $R(i, j)$, stačí jeden reprezentant $r(i, j)$. Pokud vypočítáme $r(i, j-1)$ a $r(i+1, j)$, můžeme automaticky předpokládat, že

$$r(i, j-1) \leq r(i, j) \leq r(i+1, j).$$

To omezí hledání minima: místo $j-i$ hodnot stačí prozkoumat $r(i, j-1) - r(i+1, j) + 1$ hodnot. Celkové množství (pokud $j-i = d$) je omezeno teleskopickými posloupnostmi

$$\sum d \leq j \leq ni = j - dr(i+1, j) - r(i, j-1) + 1 = r(n-d+1, n) - r(0, d-1) + n - d + 1 < 2n.$$

Časová složitost je tedy $\mathcal{O}(n^2)$.

Algoritmus a demonstrace v dodatečném materiálu.