

KMI/ASL1 – Algoritmy a složitost 1

Vyvážené stromy

Jan Konečný (já)

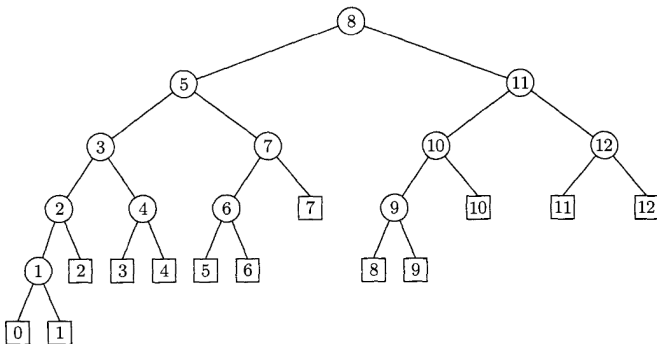
10. října 2013

Fibonacciho stromy

Fibonacciho strom řádu k má $F_{k+1} - 1$ vnitřních uzlů a $F_{k+1} + 1$ vnějších uzlů, a je konstruován následovně:

- Pokud $k = 0$ nebo $k = 1$, strom je $\boxed{0}$
- Pokud $k \geq 2$ kořen je F_k ; levý podstrom je Fibonacciho strom řádu $k - 1$; a pravý podstrom je Fibonacciho strom řádu $k - 2$ se všemi čísly zvýšenými o F_k .

Ukradený obrázek z Knutha



Vyjma vnějších uzlů se počet potomků v každém vnitřním uzlu liší čísla jejich předka o stejnou hodnotu, a tato hodnota je Fibonacciho číslo.

Například: $5 = 8 - F_4$ a $11 = 8 + F_4$. Pokud je rozdíl F_j , odpovídající rozdíl pro další větev nalevo je F_{j-1} , zatímco napravo je to F_{j-2} .

Binetův vzorec

Vyjádření Fibonachiho čísla:

$$F_n = \frac{\varphi^n - \psi^n}{\varphi - \psi} = \frac{\varphi^n - \psi^n}{\sqrt{5}},$$

kde

$$\varphi = \frac{1 + \sqrt{5}}{2} \approx 1.6180339887\dots \quad \psi = \frac{1 - \sqrt{5}}{2} \approx -0.6180339887\dots$$

Japato?

Čísla φ a ψ jsou řešení rovnic

$$x^2 = x + 1$$

$$x^n = x^{n-1} + x^{n-2},$$

takže mocniny φ a ψ splňují Fibonacciho rekurenci. To jest,

$$\varphi^n = \varphi^{n-1} + \varphi^{n-2} \quad \mathbf{a} \quad \psi^n = \psi^{n-1} + \psi^{n-2}.$$

Zjevně pro jakékoli hodnoty a, b , sekvence definovaná jako

$$U_n = a\varphi^n + b\psi^n \quad (*)$$

také splňuje stejnou rekurenci:

$$U_n = a\varphi^{n-1} + b\psi^{n-1} + a\varphi^{n-2} + b\psi^{n-2} = U_{n-1} + U_{n-2}$$

Pokud a, b určíme tak, aby $U_0 = 0, U_1 = 1$.

Výsledná posloupnost musí být Fibonacciho.

Takže potřebujeme vyřešit tuto soustavu rovnic:

$$a + b = 0$$

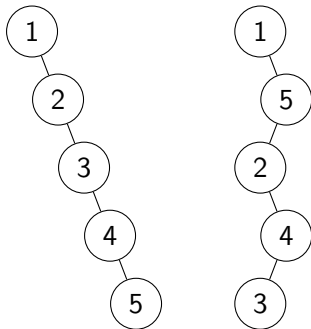
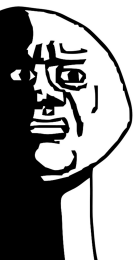
$$\varphi a + \psi b = 1$$

Řešení je $a = \frac{1}{\varphi - \psi} = \frac{1}{\sqrt{5}}, b = -a$, to se dosadí do (*) a hotovo.

Vyvážené stromy: Motivace

Z minula víme, že obyčejné vkládání do BST pro náhodná data s velkou pravděpodobností vyprodukuje dobrý BST.

Pořád je ale možnost, že nastane degenerovaný případ.



Možnosti:

- Navrhnout algoritmus, který bude udržovat strom optimální – moc složitě.
- Reorganizovat strom pokaždé, když jeho výška (vzhledem k n) překročí určitou hranici – reorganizace by nastávala příliš často: např pro hranici $5n \cdot \ln n$ by bylo potřeba zhruba $\sqrt{n/2}$ reorganizací

1962 – G.M. Adelson-Velsky, E.M. Landis: An algorithm for the organization of information

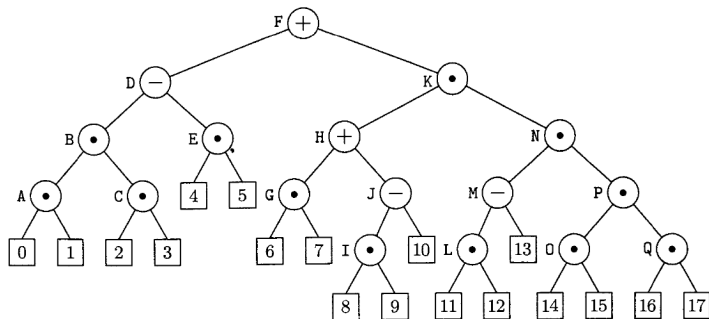
- pouze 2 bity navíc
- zajišťují čas $\mathcal{O}(\log n)$ pro vkládání i vyhledávání.

vyvážené stromy (balanced trees); též AVL-stromy (AVL-trees)

AVL-stromy

AVL strom, je BST, kde pro každý uzel platí, že výška jeho levého podstromu a výška jeho pravého podstromu se liší nejvýše o 1.
(výška h – maximální délka cesty z kořene k jakémukoli vnějšímu uzlu)

ukradený obrázek z Knutha



Každý uzel je vybavený vyvažovacím faktorem +, •, -, podle toho jestli výška pravého podstromu minus výška levého podstromu je +1, 0, -1.

Výška AVL stromu

Věta

Výška AVL-stromu s n vnitřními uzly je mezi

$$\log(n + 1) \quad a \quad 1.4404 \log(n + 2) - 0.3277.$$

- 1 BST výšky h zjevně nemůže mít více než 2^h externích uzlů; takže $n + 1 \leq 2^h$, t.j. $h \geq \lceil \log(n + 1) \rceil$ v jakémkoli BST.
- 2 Jaký je minimální počet uzlů v AVL-stromu výšky h ?

Nechť T_h je takový strom nejmenším počtem uzlů; pak

- (řekněme) levý podstrom má výšku $h - 1$,
- pravý podstrom má výšku $h - 1$ nebo $h - 2$.

Protože chceme minimum uzlů, můžeme předpokládat, že levý podstrom kořene je T_{h-1} a pravý je T_{h-2} .

Tedy Fibonacciho strom řádu $h + 1$ má nejmenší počet uzlů mezi všemi vyváženými stromy výšky h .

Taky máme, že

$$n \geq F_{h+2} - 1 \quad (1)$$

$$= \frac{\varphi^{h+2} - \psi^{h+2}}{\sqrt{5}} - 1 \quad (2)$$

$$\geq \frac{\varphi^{h+2}}{\sqrt{5}} - 2, \quad (3)$$

a tedy

$$h + 2 + \underbrace{\log_{\varphi}(1/\sqrt{5})}_c \leq \log_{\varphi}(n + 2).$$

$$h \leq \log_{\varphi}(n + 2) - c \quad (4)$$

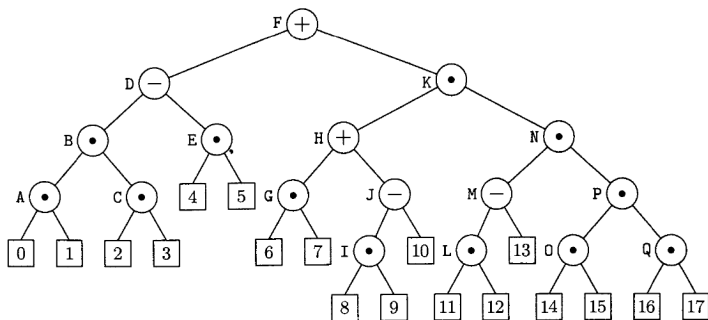
$$= \log_{\varphi}(2) \cdot \log_2(n) - c \quad (5)$$

$$= 1.44 \cdots \log_2(n) - c. \quad (6)$$

hotovo.

Vyvažování stromu

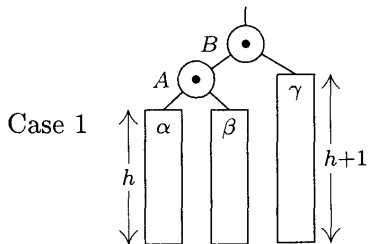
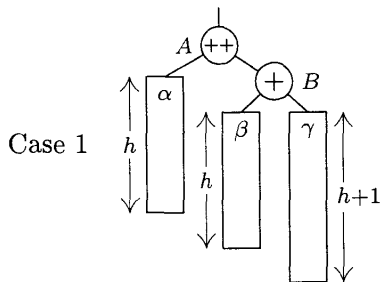
ten samý ukradený obrázek, jako předtím



Tento strom, bude vyvážený, když vložíme nový uzel (algoritmem na vkládání do BST) na místo 4,5,6,7, 10, nebo 13. Kdekoli jinde budeme potřebovat vyvážení.

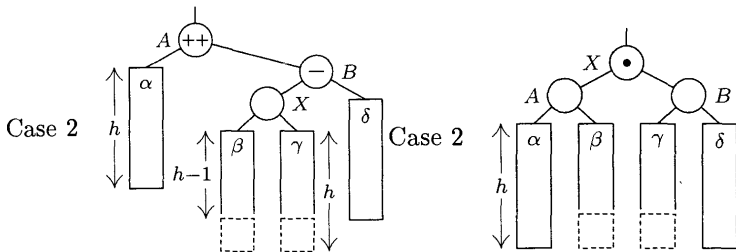
... mohou nastat dva případy:

Případ 1:



(plus zrcadlový případ)

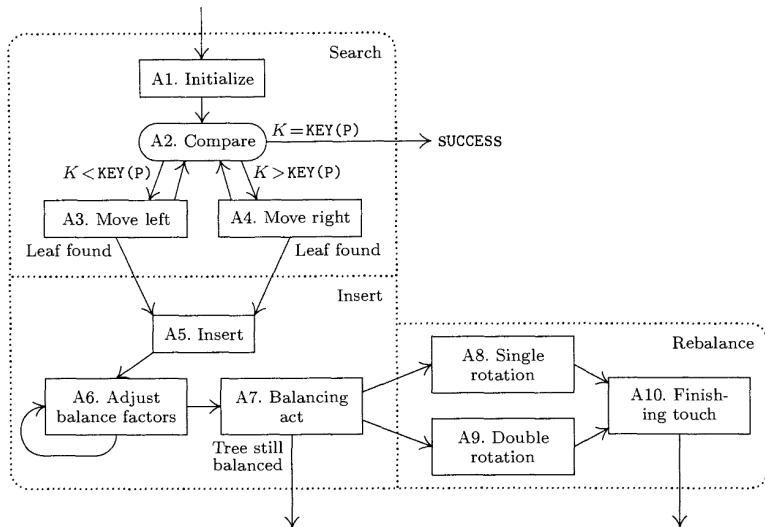
Případ 2:



(plus zrcadlový případ)

Schéma algoritmu vkládání

další ukradený obrázek



Vkládání do AVL-stromu

Algoritmus A

Vstup: neprázdný AVL-strom, klíč K

A1 [inicializace]

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

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.

A2 [porovnání]

Pokud $K < \text{key}(P)$ jdi do A3.

Pokud $K > \text{key}(P)$ jdi do A4.

Pokud $K = \text{key}(P)$ klíč už je ve stromu, končíme.

A3 [pohyb vlevo] Nastav $Q \leftarrow \text{left}(P)$.

Pokud $Q = \text{NIL}$,

nastav $Q \leftarrow \text{alloc}$,

$\text{left}(P) \leftarrow Q$ a pokračuj A5.

Jinak:

pokud $B(Q) \neq 0$, nastav $T \leftarrow P, S \leftarrow Q$.

Nastav $P \leftarrow Q$ a pokračuj A2.

A4 [pohyb vpravo]

Nastav $Q \leftarrow \text{right}(P)$.

Pokud $Q = \text{NIL}$,

nastav $Q \leftarrow \text{alloc}$,

$\text{right}(P) \leftarrow Q$ a pokračuj A5.

Jinak:

pokud $B(Q) \neq 0$, nastav $T \leftarrow P, S \leftarrow Q$.

Nastav $P \leftarrow Q$ a pokračuj A2.

A5 [vložení] $\text{key}(Q) \leftarrow K$

$\text{left}(Q) \leftarrow \text{NIL}$

$\text{right}(Q) \leftarrow \text{NIL}$

$B(Q) \leftarrow 0.$

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

Pokud $K < \text{key}(S)$, nastav $a \leftarrow -1$ jinak nastav $a \leftarrow 1$.

Nastav $R \leftarrow P \leftarrow \text{link}(a, S)$ a opakovaně dělej následující operace, dokud nenastane $P = Q$:

Pokud $K < \text{key}(P)$ nastav $B(P) \leftarrow -1, P \leftarrow \text{left}(P)$.

Pokud $K > \text{key}(P)$ nastav $B(P) \leftarrow +1, P \leftarrow \text{right}(P)$.

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

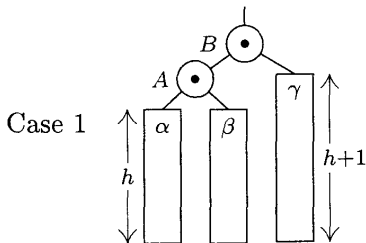
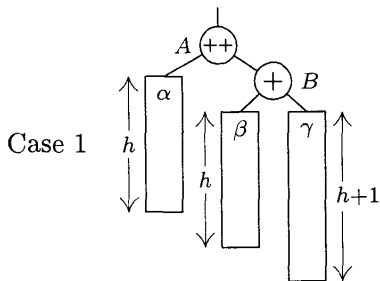
A7 [vyvažování] Může nastat několik případů:

(i) Pokud je $B(S) = 0$, zvýšila se výška, nastav $B(S) \leftarrow a$,
 $\text{left}(\text{head}) \leftarrow \text{left}(\text{head}) + 1$, skonči.

(ii) Pokud je $B(S) = -a$, strom se stal více vyvážený, nastav
 $B(S) \leftarrow 0$, skonči

(iii) Pokud je $B(S) = a$, strom není vyvážený:
pokud $B(R) = a$ pokračuj krokem A8.
pokud $B(R) = -a$ pokračuj krokem A9.

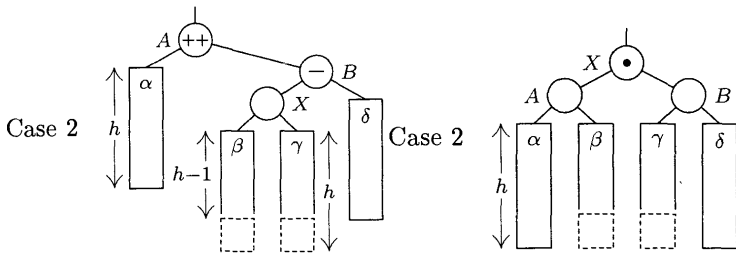
A8 [jednoduchá rotace] Nastav $P \leftarrow R$,
 $\text{link}(a, S) \leftarrow \text{link}(-a, R)$,
 $\text{link}(-a, R) \leftarrow S$,
 $B(S) \leftarrow B(R) \leftarrow 0$.
 Pokračuj krokem A10.



A9 [dvojitá rotace] Nastav $P \leftarrow \text{link}(-a, R)$, $\text{link}(-a, R) \leftarrow \text{link}(a, P)$,
 $\text{link}(a, P) \leftarrow R$, $\text{link}(a, S) \leftarrow \text{link}(-a, P)$, $\text{link}(-a, P) \leftarrow S$

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

$B(S) \leftarrow 0$. Pokračuj krokem A10.



A10 [zakončení] Pokud $S = \text{right}(T)$, nastav $\text{right}(T) \leftarrow P$ jinak nastav $\text{left}(T) \leftarrow P$

Analýza vkládání do AVL stromu

Horní hranici času v *nejhorším případě* můžeme snadno odvodit z Věty o výšce AVL-stromu.

Průměrný případ?

Žádná komplexní analýza neexistuje, na to je ten algoritmus už příliš složitý.

Pořád je tu ale dost zajímavých teoretických a empirických výsledků:

Jaký je počet B_{nh} vyvážených binárních stromů s n vnitřními uzly a výškou h ?

pro malá h ze vztahů

$$B_0(z) = 1$$

$$B_1(z) = z$$

$$B_{h+1}(z) = zB_h(z)(B_h(z) + 2B_{h-1}(z))$$

Generující posloupnost

$$B_h(z) = \sum_{n \geq 0} B_{nh} z^n$$

$$B_0(z) = 1$$

$$B_1(z) = z$$

$$B_{h+1}(z) = zB_h(z)(B_h(z) + 2B_{h-1}(z))$$

Proč:

$$\begin{aligned} b_{n,h+1} &= \sum_{i=0}^{n-1} b_{i,h-1} \cdot b_{n-i+1,h} && \text{pravý podstrom větší,} \\ &+ \sum_{i=0}^{n-1} b_{i,h} \cdot b_{n-i+1,h-1} && \text{levý podstrom větší,} \\ &+ \sum_{i=0}^{n-1} b_{i,h} b_{n-i+1,h} && \text{oba podstromy stejné.} \end{aligned}$$

$$b_{n,h+1} = \sum_{i=0}^{n-1} 2 \cdot b_{i,h-1} \cdot b_{n-i+1,h} + b_{i,h} b_{n-i+1,h}$$

$$\sum_{i=0}^{n-1} b_{n-i+1,h} (2 \cdot b_{i,h-1} + b_{i,h})$$

pak stejná finta jako na cvičení

Tedy např:

$$B_2(z) = 2z^2 + z^3$$

$$B_3(z) = 4z^4 + 6z^5 + 4z^6 + z^7$$

$$B_4(z) = 16z^7 + 32z^8 + 44z^9 + \dots + 8z^{14} + z^{15}$$

Celkový počet vyvážených stromů s výškou h je $B_h = B_h(1)$, které splňuje rekurenci:

$$B_0 = 1$$

$$B_1 = 1$$

$$B_{h+1} = B_h^2 + 2B_h B_{h-1}$$

Takže

$$B_2 = 1^2 + 2 \cdot 1 \cdot 1 = 3 \qquad = 3$$

$$B_3 = 3^2 + 2 \cdot 3 \cdot 1 = 15 \qquad = 3 \cdot 5$$

$$B_4 = 15^2 + 2 \cdot 15 \cdot 3 = 315 \qquad = 3^2 \cdot 5 \cdot 7$$

$$B_5 = \dots \qquad = 3^3 \cdot 5^2 \cdot 7 \cdot 23$$

Obecně $B_h = A_0^{F_h} A_1^{F_{h-1}} \dots A_{h-1}^{F_1} \cdot A_h^{F_0}$

kde $A_0 = 1, A_1 = 3, A_2 = 5, A_3 = 7, A_4 = 23, A_5 = 347, \dots,$

$A_h = A_{h-1} B_{h-2} + 2.$

Posloupnosti B_h a A_h rostou velmi rychle – jsou *dvojitě exponenciální*.
Lze ukázat, že existuje reálné číslo $\theta \approx 1.43687$ t.ž.

$$B_h = \lfloor \theta^{2^h} \rfloor - \lfloor \theta^{2^{h-1}} \rfloor + \lfloor \theta^{2^{h-2}} \rfloor - \dots + (-1)^h \lfloor \theta^{2^0} \rfloor$$

Pokud uvažujeme B_h stromy stejně pravděpodobné, je průměrný počet uzlů ve stromu výšky h

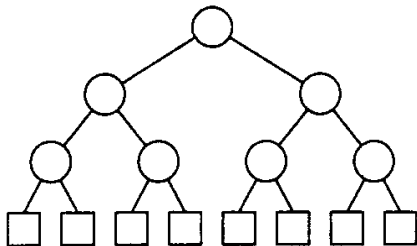
$$B'_h(1)/B_h(1) \approx (0.70118)2^h - 1. \quad (\times)$$

(důkaz jako domácí úkol)

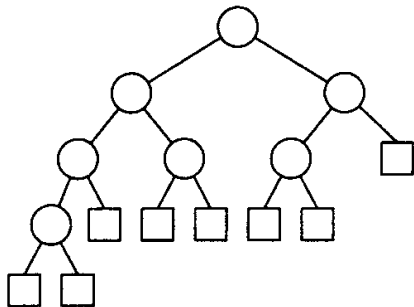
To indikuje, že výška vyváženého stromu s n uzly bude obvykle daleko blíže $\log_2 n$ než k $\log_\theta n$.

Naneštěstí tento výsledek nemá moc společného s algoritmem vkládání do AVL-stromu. Při jeho použití vznikají některé stromy se značně vyšší pravděpodobností, než jiné:

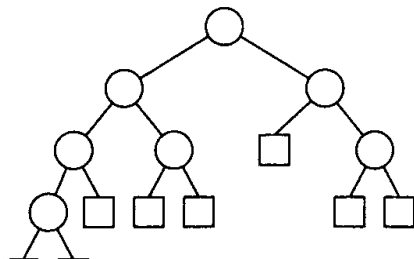
Např. uvažme případ $n = 7$, kde existuje 17 možných AVL-stromů, a $7! = 5040$ možných pořadí, ve kterých lze vložit 7 klíčů. perfektně vyvážený strom obdržíme v 2160 případech:



Fibonachiho strom obdržíme v 144 případech:



podobný strom obdržíme v 216 případech:



Fakt, že perfektně vyvážený strom vznikne vysokou pravděpodobností společně se vztahem (x) .

Z toho bychom mohli usuzovat, že průměrně bude pro hledání v AVL potřeba okolo $\log n + c$ porovnání.

Ovšem koeficient u termu $\log n$ nebude přesně 1, protože kořen stromu by pak byl blízko mediánu a kořeny jeho podstromu blízko kvartilům.

Jednoduchá nebo dvojitá rotace by nemohla jednoduše udržet kořen blízku mediánu.

Empirické testy ukazují, že průměrný počet porovnání pro vložení n -tého prvku je asi $1.01 \log n + 0.1$, vyjma případů, kdy n je malé.

DÚ: Ověřte.

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.

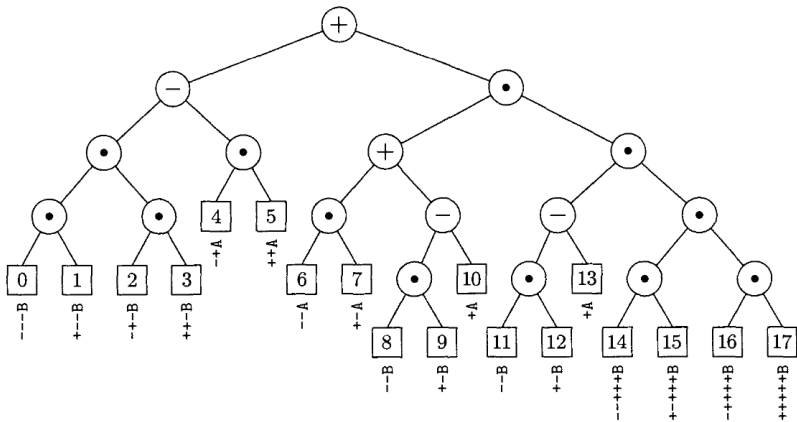
Píšeme tuto sekvenci dokud nenarazíme na uzel s nenulovým vyvažovacím faktorem, nebo na kořen.

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

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)