

# Datové struktury I

## 2. přednáška: Splay stromy

Jirka Fink

<https://ktiml.mff.cuni.cz/~fink/>

Katedra teoretické informatiky a matematické logiky  
Matematicko-fyzikální fakulta  
Univerzita Karlova v Praze

Zimní semestr 2026/27

Licence: Creative Commons BY-NC-SA 4.0

## Cíl

Pro danou posloupnost operací FIND najít binární vyhledávací strom minimalizující celkovou dobu vyhledávání.

## Cíl

Pro danou posloupnost operací FIND najít binární vyhledávací strom minimalizující celkovou dobu vyhledávání.

## Formálně

Máme prvky  $x_1, \dots, x_n$  s váhami  $w_1, \dots, w_n$ . Cena stromu je  $\sum_{i=1}^n w_i h_i$ , kde  $h_i$  je hloubka prvku  $x_i$ . Staticky optimální strom je binární vyhledávací strom s minimální cenou.

## Cíl

Pro danou posloupnost operací FIND najít binární vyhledávací strom minimalizující celkovou dobu vyhledávání.

## Formálně

Máme prvky  $x_1, \dots, x_n$  s váhami  $w_1, \dots, w_n$ . Cena stromu je  $\sum_{i=1}^n w_i h_i$ , kde  $h_i$  je hloubka prvku  $x_i$ . Staticky optimální strom je binární vyhledávací strom s minimální cenou.

## Konstrukce (cvičení)

- $\mathcal{O}(n^3)$  – triviálně dynamickým programováním
- $\mathcal{O}(n^2)$  – vylepšené dynamické programování (Knuth, 1971)

## Cíl

Pro danou posloupnost operací FIND najít binární vyhledávací strom minimalizující celkovou dobu vyhledávání.

## Formálně

Máme prvky  $x_1, \dots, x_n$  s váhami  $w_1, \dots, w_n$ . Cena stromu je  $\sum_{i=1}^n w_i h_i$ , kde  $h_i$  je hloubka prvku  $x_i$ . Staticky optimální strom je binární vyhledávací strom s minimální cenou.

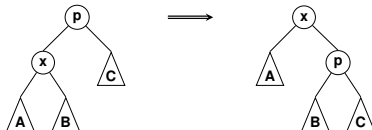
## Konstrukce (cvičení)

- $\mathcal{O}(n^3)$  – triviálně dynamickým programováním
- $\mathcal{O}(n^2)$  – vylepšené dynamické programování (Knuth, 1971)

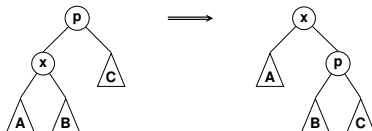
## Jak postupovat, když neznáme váhy předem?

- Pomocí rotací budeme udržovat často vyhledávané prvky blízko kořene
- Operací SPLAY „rotujeme“ zadaný prvek až ke kořeni
- Operace FIND vždy volá SPLAY na hledaný prvek

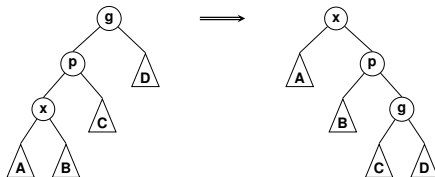
- Zig rotace: Otec  $p$  prvku  $x$  je kořen



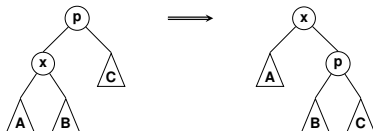
- Zig rotace: Otec  $p$  prvku  $x$  je kořen



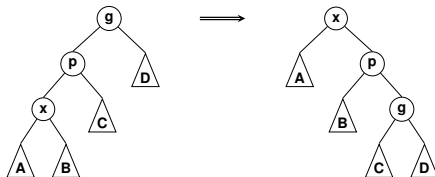
- Zig-zig rotace:  $x$  a  $p$  jsou oba pravými nebo oba levými syny



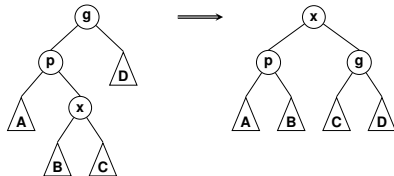
- Zig rotace: Otec  $p$  prvku  $x$  je kořen



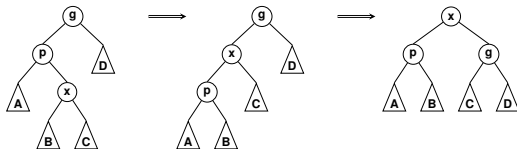
- Zig-zig rotace:  $x$  a  $p$  jsou oba pravými nebo oba levými syny



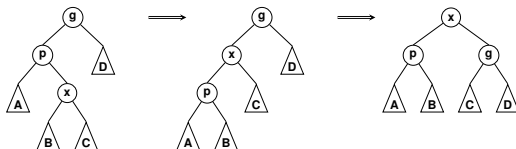
- Zig-zag rotace:  $x$  je pravý syn a  $p$  je levý syn nebo opačně



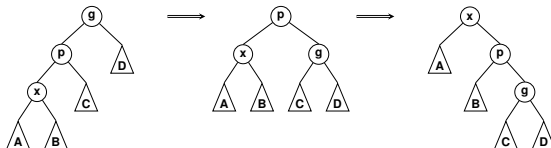
- Zig-zag rotace jsou pouze dvě jednoduché rotace prvku  $x$  s aktuálním otcem



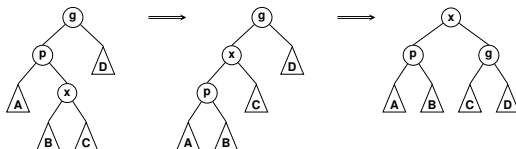
- Zig-zag rotace jsou pouze dvě jednoduché rotace prvku  $x$  s aktuálním otcem



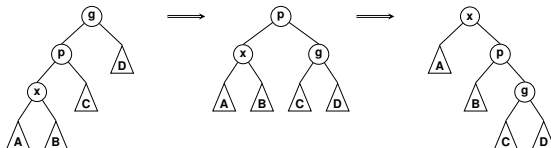
- Zig-zig rotace jsou také dvě rotace,



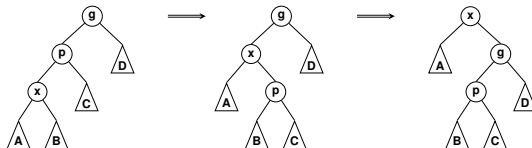
- Zig-zag rotace jsou pouze dvě jednoduché rotace prvku x s aktuálním otcem



- Zig-zig rotace jsou také dvě rotace,



- ale dvě rotace prvku x s aktuálním otcem by vedly ke špatnému výsledku



## Lemma

Pro  $a, b, c \in \mathbb{R}^+$  splňující  $a + b \leq c$  platí  $\log_2(a) + \log_2(b) \leq 2 \log_2(c) - 2$ .

## Lemma

Pro  $a, b, c \in \mathbb{R}^+$  splňující  $a + b \leq c$  platí  $\log_2(a) + \log_2(b) \leq 2 \log_2(c) - 2$ .

## Důkaz

- Platí  $4ab = (a + b)^2 - (a - b)^2$
- Z nerovností  $(a - b)^2 \geq 0$  a  $a + b \leq c$  plyne  $4ab \leq c^2$
- Zlogaritmováním dostáváme  $\log_2(4) + \log_2(a) + \log_2(b) \leq \log_2(c^2)$

## Lemma

Pro  $a, b, c \in \mathbb{R}^+$  splňující  $a + b \leq c$  platí  $\log_2(a) + \log_2(b) \leq 2 \log_2(c) - 2$ .

## Důkaz

- Platí  $4ab = (a + b)^2 - (a - b)^2$
- Z nerovností  $(a - b)^2 \geq 0$  a  $a + b \leq c$  plyne  $4ab \leq c^2$
- Zlogaritmováním dostáváme  $\log_2(4) + \log_2(a) + \log_2(b) \leq \log_2(c^2)$

## Značení

- Nechť velikost  $s(x)$  je počet vrcholů v podstromu  $x$  (včetně vrcholu  $x$ )
- Potenciál vrcholu  $x$  je  $\Phi(x) = \log_2(s(x))$
- Potenciál  $\Phi$  stromu je součet potenciálů všech vrcholů

## Lemma

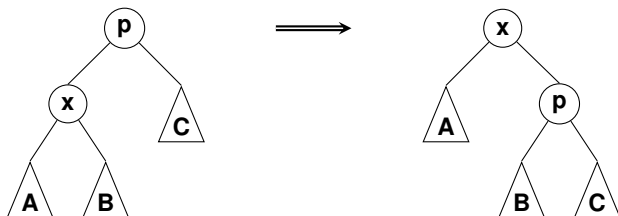
Pro  $a, b, c \in \mathbb{R}^+$  splňující  $a + b \leq c$  platí  $\log_2(a) + \log_2(b) \leq 2 \log_2(c) - 2$ .

## Důkaz

- Platí  $4ab = (a + b)^2 - (a - b)^2$
- Z nerovností  $(a - b)^2 \geq 0$  a  $a + b \leq c$  plyne  $4ab \leq c^2$
- Zlogaritmováním dostáváme  $\log_2(4) + \log_2(a) + \log_2(b) \leq \log_2(c^2)$

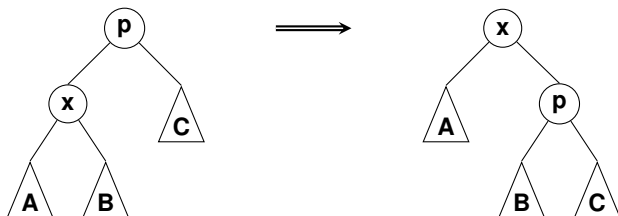
## Značení

- Nechť velikost  $s(x)$  je počet vrcholů v podstromu  $x$  (včetně vrcholu  $x$ )
- Potenciál vrcholu  $x$  je  $\Phi(x) = \log_2(s(x))$
- Potenciál  $\Phi$  stromu je součet potenciálů všech vrcholů
- $s'$  a  $\Phi'$  jsou velikosti a potenciály po jedné rotaci
- Předpokládáme, že jednoduchou rotaci zvládneme v jednotkovém čase



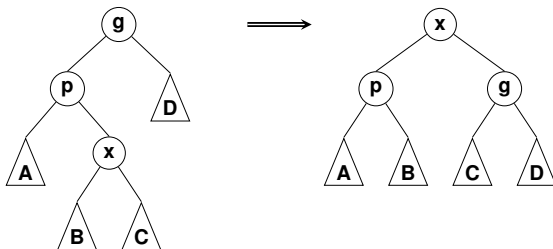
## Analýza

- $\Phi'(x) = \Phi(p)$
- $\Phi'(p) < \Phi'(x)$
- $\Phi'(u) = \Phi(u)$  pro všechny ostatní vrcholy  $u$



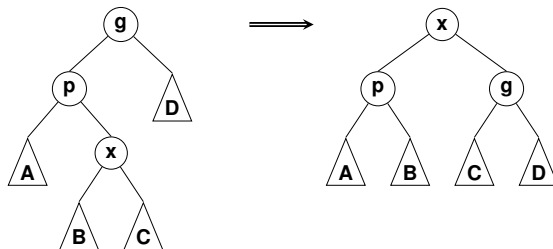
## Analýza

- $\Phi'(x) = \Phi(p)$
- $\Phi'(p) < \Phi'(x)$
- $\Phi'(u) = \Phi(u)$  pro všechny ostatní vrcholy  $u$
- $\Phi' - \Phi = \sum_u (\Phi'(u) - \Phi(u))$   
 $= \Phi'(p) - \Phi(p) + \Phi'(x) - \Phi(x)$   
 $\leq \Phi'(x) - \Phi(x)$



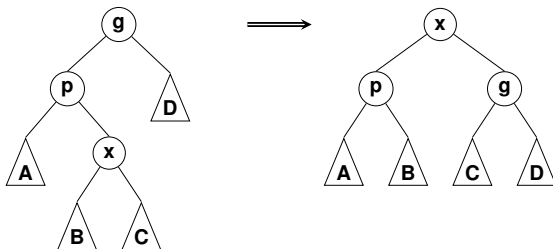
## Analýza

- 1  $\Phi'(x) = \Phi(g)$
- 2  $\Phi(x) < \Phi(p)$



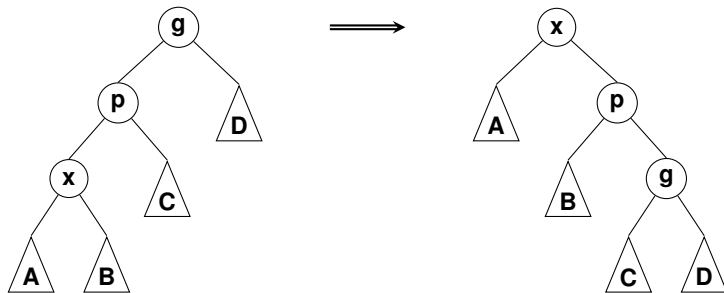
## Analýza

- 1  $\Phi'(x) = \Phi(g)$
- 2  $\Phi(x) < \Phi(p)$
- 3  $\Phi'(p) + \Phi'(g) \leq 2\Phi'(x) - 2$ 
  - $s'(p) + s'(g) \leq s'(x)$
  - Z lematu plyne  $\log_2(s'(p)) + \log_2(s'(g)) \leq 2\log_2(s'(x)) - 2$



## Analýza

- 1  $\Phi'(x) = \Phi(g)$
- 2  $\Phi(x) < \Phi(p)$
- 3  $\Phi'(p) + \Phi'(g) \leq 2\Phi'(x) - 2$ 
  - $s'(p) + s'(g) \leq s'(x)$
  - Z lematu plyne  $\log_2(s'(p)) + \log_2(s'(g)) \leq 2\log_2(s'(x)) - 2$
- 4  $\Phi' - \Phi = \Phi'(g) - \Phi(g) + \Phi'(p) - \Phi(p) + \Phi'(x) - \Phi(x) \leq 2(\Phi'(x) - \Phi(x)) - 2$



## Analýza

- $\Phi'(x) = \Phi(g)$
- $\Phi(x) < \Phi(p)$
- $\Phi'(p) < \Phi'(x)$
- $s(x) + s'(g) \leq s'(x)$
- $\Phi(x) + \Phi'(g) \leq 2\Phi'(x) - 2$
- $\Phi' - \Phi = \Phi'(g) - \Phi(g) + \Phi'(p) - \Phi(p) + \Phi'(x) - \Phi(x) \leq 3(\Phi'(x) - \Phi(x)) - 2$

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:

$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:

$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$

- Amortizovaný čas jedné zig rotace:

$$T + \Phi' - \Phi \leq 1 + \Phi'(x) - \Phi(x) \leq 1 + 3(\Phi'(x) - \Phi(x))$$

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:  
$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$
- Amortizovaný čas jedné zig rotace:  
$$T + \Phi' - \Phi \leq 1 + \Phi'(x) - \Phi(x) \leq 1 + 3(\Phi'(x) - \Phi(x))$$
- Nechť  $\Phi_i$  je potenciál po  $i$ -tém kroku a  $T_i$  je skutečný čas  $i$ -tého kroku

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:  
$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$
- Amortizovaný čas jedné zig rotace:  
$$T + \Phi' - \Phi \leq 1 + \Phi'(x) - \Phi(x) \leq 1 + 3(\Phi'(x) - \Phi(x))$$
- Nechť  $\Phi_i$  je potenciál po  $i$ -tém kroku a  $T_i$  je skutečný čas  $i$ -tého kroku
- Amortizovaný čas (počet jednoduchých rotací) jedné operace SPLAY:

$$\begin{aligned} \sum_{i\text{-tá rotace}} (T_i + \Phi_i - \Phi_{i-1}) &\leq 1 + \sum_{i\text{-tá rotace}} 3(\Phi_i(x) - \Phi_{i-1}(x)) \\ &\leq 1 + 3(\Phi_{\text{konec}}(x) - \Phi_0(x)) \\ &\leq 1 + 3 \log_2 n = \mathcal{O}(\log n) \end{aligned}$$

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:  
$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$
- Amortizovaný čas jedné zig rotace:  
$$T + \Phi' - \Phi \leq 1 + \Phi'(x) - \Phi(x) \leq 1 + 3(\Phi'(x) - \Phi(x))$$
- Nechť  $\Phi_i$  je potenciál po  $i$ -tém kroku a  $T_i$  je skutečný čas  $i$ -tého kroku
- Amortizovaný čas (počet jednoduchých rotací) jedné operace SPLAY:

$$\sum_{i\text{-tá rotace}} (T_i + \Phi_i - \Phi_{i-1}) \leq 1 + \sum_{i\text{-tá rotace}} 3(\Phi_i(x) - \Phi_{i-1}(x))$$

$$\leq 1 + 3(\Phi_{\text{konec}}(x) - \Phi_0(x))$$

$$\leq 1 + 3 \log_2 n = \mathcal{O}(\log n)$$

- Amortizovaný čas jedné operace SPLAY je  $\mathcal{O}(\log n)$

## Amortizovaný čas

- Amortizovaný čas jedné zigzig nebo zigzag rotace:  
$$T + \Phi' - \Phi \leq 2 + 3(\Phi'(x) - \Phi(x)) - 2 = 3(\Phi'(x) - \Phi(x))$$
- Amortizovaný čas jedné zig rotace:  
$$T + \Phi' - \Phi \leq 1 + \Phi'(x) - \Phi(x) \leq 1 + 3(\Phi'(x) - \Phi(x))$$
- Nechť  $\Phi_i$  je potenciál po  $i$ -tém kroku a  $T_i$  je skutečný čas  $i$ -tého kroku
- Amortizovaný čas (počet jednoduchých rotací) jedné operace SPLAY:

$$\begin{aligned} \sum_{i\text{-tá rotace}} (T_i + \Phi_i - \Phi_{i-1}) &\leq 1 + \sum_{i\text{-tá rotace}} 3(\Phi_i(x) - \Phi_{i-1}(x)) \\ &\leq 1 + 3(\Phi_{\text{konec}}(x) - \Phi_0(x)) \\ &\leq 1 + 3 \log_2 n = \mathcal{O}(\log n) \end{aligned}$$

- Amortizovaný čas jedné operace SPLAY je  $\mathcal{O}(\log n)$

## Skutečný čas $k$ operací SPLAY

- Potenciál vždy splňuje  $0 \leq \Phi \leq n \log_2 n$
- Rozdíl mezi konečným a počátečním potenciálem je nejvýše  $n \log_2 n$
- Celkový čas  $k$  operací SPLAY je  $\mathcal{O}((n + k) \log n)$

## Postup

- 1 Vložit/smazat daný vrchol stejně jako v BVS
- 2 SPLAY nejhlubšího navštíveného vrcholu

## Postup

- 1 Vložit/smazat daný vrchol stejně jako v BVS
- 2 SPLAY nejhlubšího navštíveného vrcholu

## Amortizovaná složitost

- Nalezení potřebných vrcholů má stejnou složitost jako SPLAY:  $\mathcal{O}(\log n)$
- Vlastní smazání vrcholu trvá  $\mathcal{O}(1)$  a potenciál stromu klesne
- Přidáním listu se potenciál vrcholů  $u_1, \dots, u_h$  na cestě do kořene zvýší o  $\sum_{i=1}^h \log(s(u_i) + 1) - \log(s(u_i)) \leq \log(n)$
- Amortizovaná složitost operací INSERT a DELETE je  $\mathcal{O}(\log n)$

## Věta

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$  a necht'  $z_i$  je počet různých prvků vyhledaných před předchozím vyhledáním prvku  $x_i$ . Celková složitost vyhledání prvků  $x_1, \dots, x_k$  je  $\mathcal{O}(n \log n + k + \sum_i \log(1 + z_i))$ .

## Věta

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$  a necht'  $z_i$  je počet různých prvků vyhledaných před předchozím vyhledáním prvku  $x_i$ . Celková složitost vyhledání prvků  $x_1, \dots, x_k$  je  $\mathcal{O}(n \log n + k + \sum_i \log(1 + z_i))$ .

## Věta (vyhledávání podmnožiny prvků)

Jestliže provedeme  $k$  vyhledání prvků z podmnožiny velikosti  $m$ , pak celková složitost je  $\mathcal{O}(n \log n + k + k \log m)$

## Věta

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$  a necht'  $z_i$  je počet různých prvků vyhledaných před předchozím vyhledáním prvku  $x_i$ . Celková složitost vyhledání prvků  $x_1, \dots, x_k$  je  $\mathcal{O}(n \log n + k + \sum_i \log(1 + z_i))$ .

## Věta (vyhledávání podmnožiny prvků)

Jestliže provedeme  $k$  vyhledání prvků z podmnožiny velikosti  $m$ , pak celková složitost je  $\mathcal{O}(n \log n + k + k \log m)$

## Věta (vyhledávání prvků v rostoucím pořadí)

Jestliže posloupnost vyhledávání  $S$  obsahuje prvky v rostoucím pořadí, tak celkový čas na vyhledávání  $S$  ve splay stromu je  $\mathcal{O}(n)$ .

## Otázka

- Existuje posloupnost vyhledání, na které je splay strom asymptoticky rychlejší než staticky optimální strom zkonstruovaný pro tuto posloupnost?
- Existuje posloupnost vyhledání, na které je staticky optimální strom asymptoticky rychlejší než splay strom?

## Otázka

- Existuje posloupnost vyhledání, na které je splay strom asymptoticky rychlejší než staticky optimální strom zkonstruovaný pro tuto posloupnost?
- Existuje posloupnost vyhledání, na které je staticky optimální strom asymptoticky rychlejší než splay strom?

## Věta (statická optimalita)

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$ , kde každý prvek  $x$  je vyhledán aspoň jednou. Nechť  $T$  je statický strom na  $X$  a složitost vyhledání prvků v posloupnosti je  $f$ . Pak složitost vyhledání ve splay stromu je  $\mathcal{O}(f)$ .

## Otázka

- Existuje posloupnost vyhledání, na které je splay strom asymptoticky rychlejší než staticky optimální strom zkonstruovaný pro tuto posloupnost?
- Existuje posloupnost vyhledání, na které je staticky optimální strom asymptoticky rychlejší než splay strom?

## Věta (statická optimalita)

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$ , kde každý prvek  $x$  je vyhledán aspoň jednou. Nechť  $T$  je statický strom na  $X$  a složitost vyhledání prvků v posloupnosti je  $f$ . Pak složitost vyhledání ve splay stromu je  $\mathcal{O}(f)$ .

## Dynamická optimalita

- Jsou dvojité rotace ve splay stromu nejlepší možné?

## Otázka

- Existuje posloupnost vyhledání, na které je splay strom asymptoticky rychlejší než staticky optimální strom zkonstruovaný pro tuto posloupnost?
- Existuje posloupnost vyhledání, na které je staticky optimální strom asymptoticky rychlejší než splay strom?

## Věta (statická optimalita)

Nechť  $x_1, \dots, x_k$  je posloupnost vyhledávání prvků z množiny  $X$ , kde každý prvek  $x$  je vyhledán aspoň jednou. Nechť  $T$  je statický strom na  $X$  a složitost vyhledání prvků v posloupnosti je  $f$ . Pak složitost vyhledání ve splay stromu je  $\mathcal{O}(f)$ .

## Dynamická optimalita

- Jsou dvojité rotace ve splay stromu nejlepší možné?
- Existuje dynamický binární vyhledávací strom, který by byl asymptoticky lepší než splay strom?

## Výhody a nevýhody Splay stromů

- + Nepotřebuje paměť na speciální příznaky
- + Efektivně využívají procesorové cache (Temporal locality)
- Rotace zpomalují vyhledávání
- Vyhledávání nelze jednoduše paralelizovat
- Výška stromu může být i lineární

## Výhody a nevýhody Splay stromů

- + Nepotřebuje paměť na speciální příznaky
- + Efektivně využívají procesorové cache (Temporal locality)
  - Rotace zpomalují vyhledávání
  - Vyhledávání nelze jednoduše paralelizovat
  - Výška stromu může být i lineární

## Aplikace

- Cache, virtuální paměť, sítě, file system, komprese dat, ...
- Windows, gcc compiler and GNU C++ library, sed string editor, Fore Systems network routers, Unix malloc, Linux loadable kernel modules, ...

- (a,b)-stromy
- Souvislost (2,4)-stromů a červeno-černých stromů