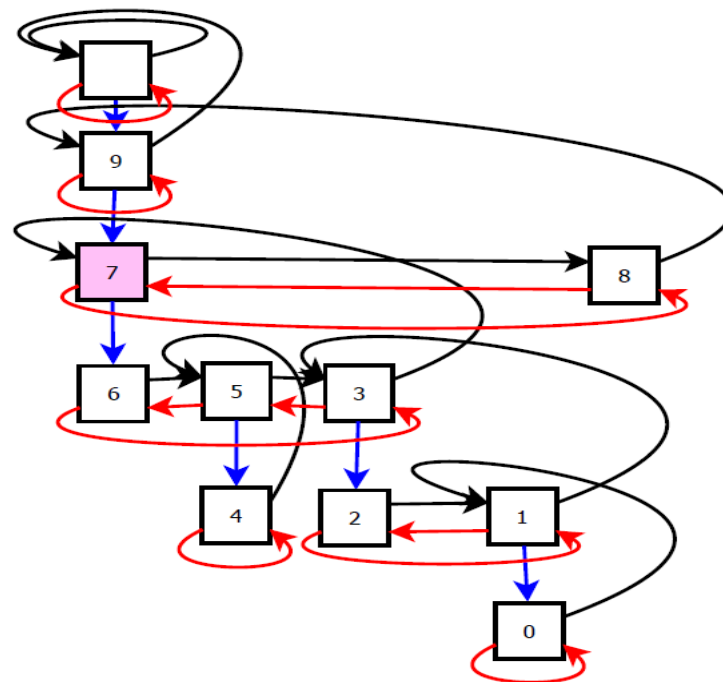
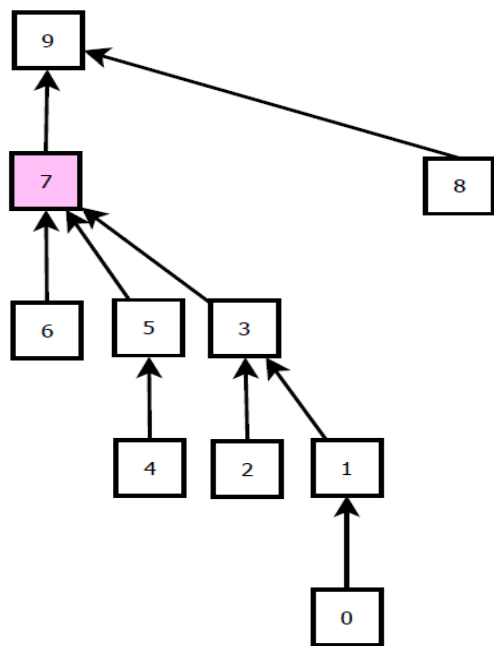


Padovan heaps

Vladan Majerech



Základní princip – velmi drahé porovnávání

- Porovnááme jen když musíme (v podstatě veškerou práci dělá FindMin)
- Výsledky porovnávání nezhazujeme
- Nesmíme organizací porovnávání ztratit řádově víc času než porovnáním

Invariant c-q úzkosti

Každý prvek haldy bude mít definovaný řád ($rank_v$) (nový prvek 0).

- Pro $c > 0$ a $1 < q \leq 2$ platí:

Libovolný vrchol řádu k má alespoň cq^k velký podstrom předchůdců.

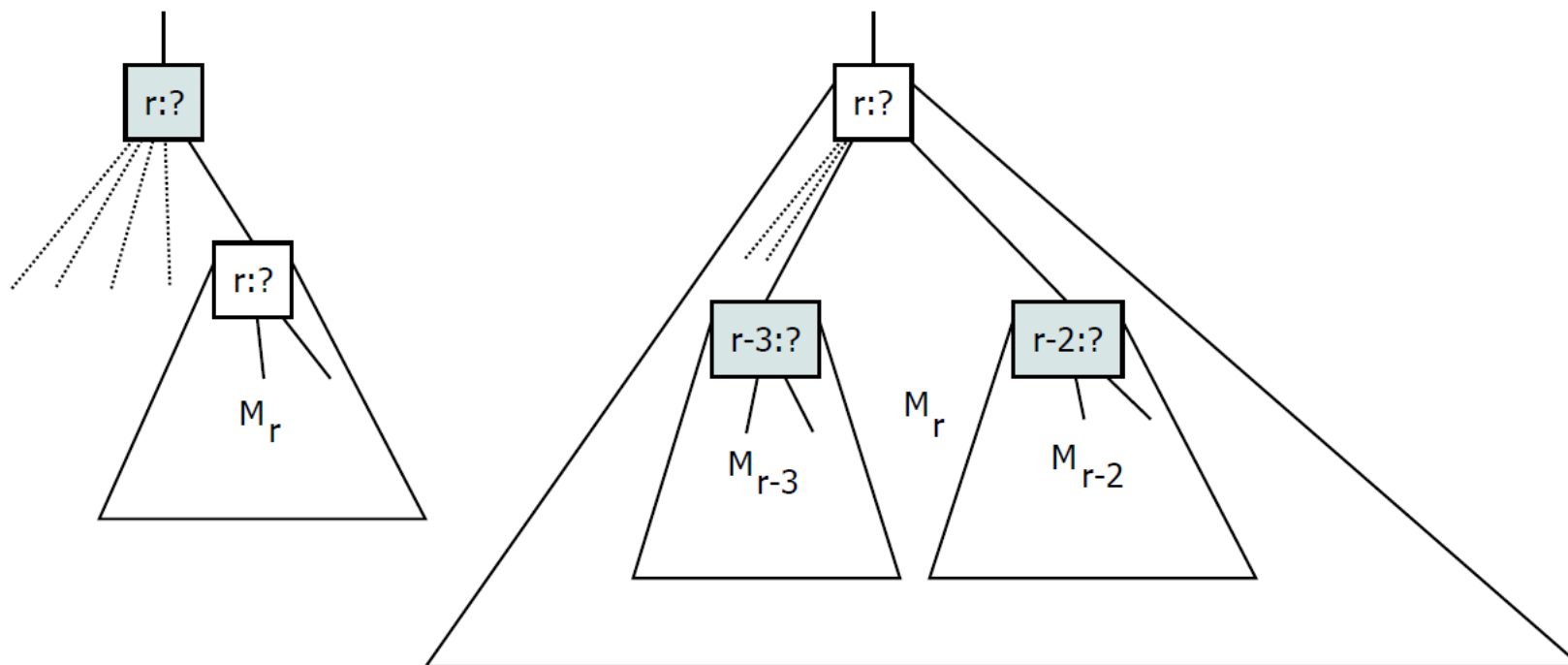
Ušetření pointeru

- Kvůli Decrement Fibonacciho haldy potřebují pointer na následníka prvku, abychom mohli propagovat snižování řádů.
- Šlo by tento pointer ušetřit a použít odkaz jen v posledním předchůdci (místo odkazu na dalšího)?
- Pak bychom nemohli propagovat snižování řádu s výjimkou prvků na konci seznamu, které mají k následníkovi $O(1)$ přístup.
- Propagaci provádíme jen v posledních dvou prvcích seznamu (předchůdcích největšího řádu).
- Worst case čas DeleteMin klesne zpět na $O(1)$.

rank_v a *wrank_v*

- Pro černý vrchol je $wrank = 1 + rank$, pro bílý vrchol je $wrank = rank$, pro null (chybějící vrchol) je $wrank = -1$.
- Seznam předchůdců (dětí) má všechny červené nalevo, pak neklesá $wrank$ (pokud je definovaný).
- Vezmeme z bílých a černých přímých předchůdců (dětí) vrcholu v dva nejpravější ... zprava w_0 a w_1 . (mohou chybět)
 - S. Bezpečný: $wrank_{w_1} + 1 = wrank_{w_0}$; $rank_v = wrank_{w_0} + 1$.
 - D. Nebezpečný: $wrank_{w_1} + 1 < wrank_{w_0}$, a w_0 je bílý: $rank_v = wrank_{w_0}$. (v nemůže být bílé)
 - F. Zakázaný: $wrank_{w_1} + 1 < wrank_{w_0}$, a w_0 je černý: zažluť w_0 a přepočítej.

Rekurence minimálních stromů



Úzkost

- Pro minimální velikost M_k stromu řádu k máme

$$M_k = 1 + M_{k-2} + M_{k-3}.$$

- Úzká souvislost s Padovan posloupností a tedy c-q úzkost pro

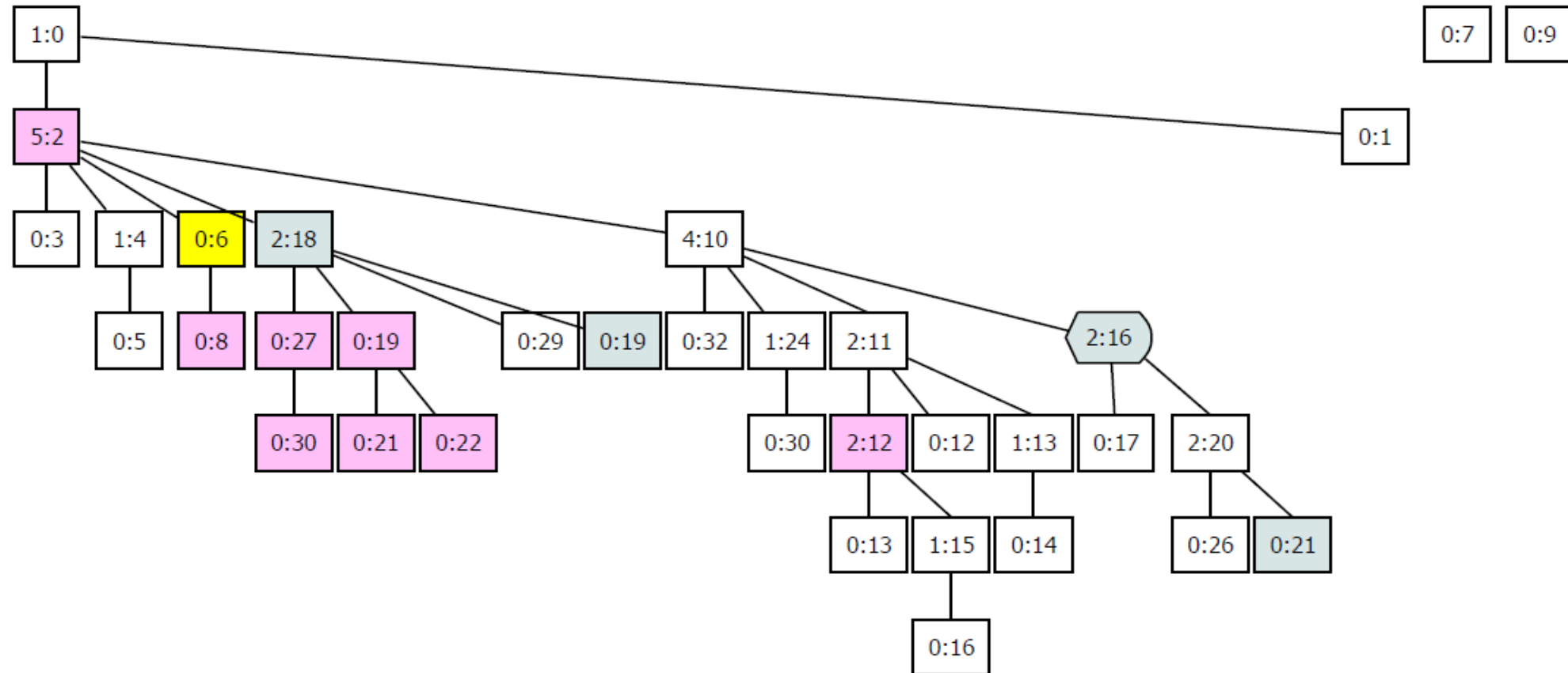
$$q = \sqrt[3]{\left(\frac{1}{2}\left(1 + \sqrt{\frac{23}{27}}\right)\right)} + \sqrt[3]{\left(\frac{1}{2}\left(1 - \sqrt{\frac{23}{27}}\right)\right)} \approx 1.324718.$$

- $(q^3 = q + 1)$

Decrement zcela vybarvený

- Utrhneme prvek (kandidát na minimum), a pokud byl jedním z dvou nejpravějších předchůdců, vyvoláme kaskádovou aktualizaci řádů na jeho následníkovi.
- Při kaskádové aktualizaci vždy přepočítáme rank. Pokud nedošlo ke změně, výpočet končí.
- Při snížení ranku bílého prvku o 1 prvek začerníme.
- Pokud došlo ke snížení ranku černého prvku, je obarven žlutě, stejně tak při snížení ranku bílého prvku aspoň o 2.
- Je-li prvek mezi pravými dvěmi *i*, pokračuje kaskádové přepočítávání.
- Při přepočítávání ranku prvku nejprve žluté vrcholy mezi pravými dvěmi *i* přebarvujeme na červeně a přemísťujeme na levý kraj seznamu. Pokud narazíme na červený prvek, víme, že další bílý či černý prvek neexistuje.

Obrázek se všemi barvami (klíče mají mínus)



Shrnutí

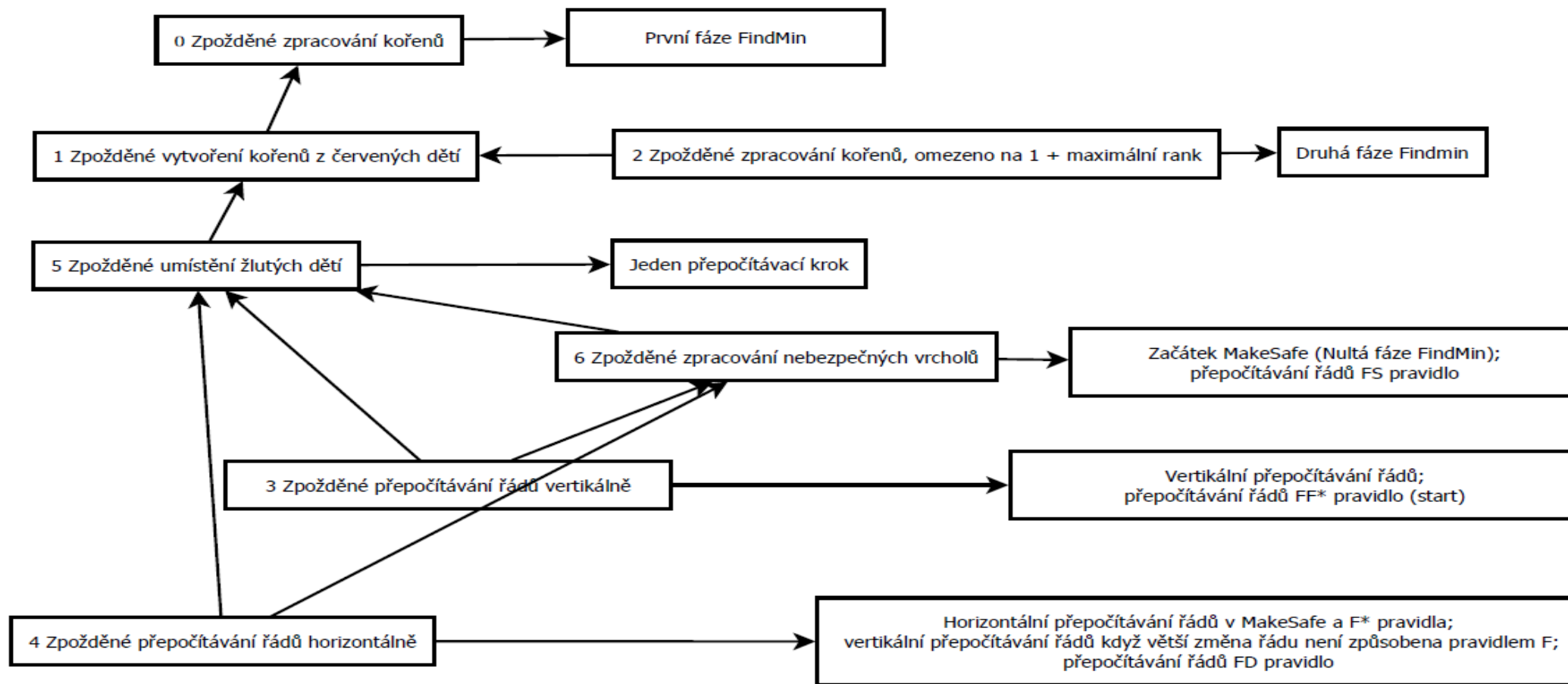
- Potenciál vůči němuž je analýza je tedy

$$\Phi_0 t_0 + \Phi_1 t_1 + \Phi_2 t_2 + 2\Phi_3 t_3 + \Phi_4 t_4 + \Phi_5 t_5 + \Phi_6 t_6$$

kde $t_0 \leq t_1 < t_2, t_1 < t_5 < t_6, t_5 + t_6 < t_3, t_5 + t_6 < t_4$ jsou vhodné konstanty.

- Φ_0 je počet kandidátů.
- Φ_2 je počet kandidátů, nejvýše však nejvyšší dosažitelný řád + 1.
- Φ_1, Φ_3 , resp. Φ_5 je počet všech červených, černých, resp. žlutých předchůdců.
- Φ_4 je součet rozdílů mezi rankem a počtem bílých a černých přímých předků.
- Φ_6 je počet nebezpečných vrcholů

Schéma placení



Soutěž struktur

- Obhájce jedné struktury vygeneruje posloupnost volání metod, obě funkce provedou danou posloupnost a změří se poměr jejich časů.
- Haldy dle principu velmi drahého porovnávání vítězí nad standardními haldami hodnotou $\theta(\log n)$ pro posloupnost
i=0;Repeat (Insert(-i), Insert(-(i+1)), FindMin, DeleteMin, i++),
protože maximální řád, který vznikne bude 4.
- Standardní haldy mohou vynutit jedině poměr $\theta(1)$.