3. cvičení

Datové struktury I, 16. 10. 2025 https://iuuk.mff.cuni.cz/~chmel/2526/ds1/

Úloha 1 (Splay stromy mají potenciál)
Ujistěte se, že chápete, jak je definovaný potenciál ve splay stromech a že rozumíte hlavním myšlenkám analýzy amortizované složitosti splaye.

a) Jak je definován potenciál splay stromu?

b) Jaký je potenciál perfektně vyváženého stromu? (Stačí nám rozumný horní a dolní odhad, pro jednoduchost předpokládejte $n = 2^k - 1$.)

c) Jaký je potenciál cesty? (Opět stačí rozumné odhady.)

d) Jaká je amortizovaná cena rotace (dvojité a jednoduché)?

e) Jaká je amortizovaná cena celého splaye a jak plyne z amortizovaných cen rotace?

f) Jaká je *reálná* cena celého splaye (a v jakých jednotkách ji vlastně počítáme)?

Řešení a) Označíme si $T(v)$ podstrom s kořenem v, velikost $s(v) = |T(v)|$, hodnost $r(v) = \log_2(s(v))$, potenciál pak je $\Phi = \sum_{v \in V} r(v)$, a (tradičně) značíme n jako počet vrcholů ve stromě.

b) Dvojitá $3r'(x) - 3r(x)$, kde r' je rank po, r je rank před, a jednoduchá $3r'(x) - 3r(x) + 1$.

c) Nejvýše $3r'(x) - 3r(x) + 1 \in \mathcal{O}(\log n)$ - tohle vyjde, protože součty amortizovaných cen rotací teleskopují.

d) Počet provedených rotací - v rotacích.

e) $\sum_{i=0}^{k-1} 2^i \cdot (\log_2(2^{k-i} - 1)) \leq \sum_{i=0}^{k-1} 2^i \cdot (\log_2(2^{k-i})) = \sum_{i=0}^{k-1} 2^i \cdot (k-i) = \sum_{\ell=1}^{k} \sum_{i=0}^{\ell-1} 2^i = \sum_{\ell=1}^{k} 2^\ell - 1 = 2^{k+1} - 1 - (k+1) = 2^{k+1} - k - 2 = 2(2^k - 1) - k = 2n - \log n$. Zároveň máme triviální spodní odhad $n/4$, protože úplný binární strom má $2^{k-2} \approx n/4$ vrcholů v úrovni těsně nad listy.

f) $\sum_{i=1}^{n} \log(i) = \log(n!) \geq \frac{n}{2} \log\left(\frac{n}{2}\right)$, zároveň můžeme odhadnout $\log(n!) \leq n \log n$, protože $n! \leq n^n$.

Úloha 2 (Semisplay)
V krocích LL a PP (neboli zig-zig) můžeme být línější a místo provedení dvojrotace provést jen jednoduchou rotaci vrchní hrany. Konkrétně v případě LL mějme tři vrcholy x, y a z takové, že x je levým synem y a y je levým synem z. Pak při splayování x provedeme jednoduchou rotaci hrany zy a dále pokračujeme ve splayování vrcholu y (místo x); původně splayovaný prvek se tedy *nemusí dostat* do kořene. Kroky LP a PL (zig-zag) nebo závěrečná jednoduchá rotace fungují stejně. Zanalyzujte amortizovaná cenu operace semisplay.

Řešení
Provedeme stejnou analýzu jako u standardního splaye, uvědomíme si, že u zig-zagu nám vlastně stačil odhad $2 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z) \leq 2 + r'(y) + r'(z) - 2r(x) \leq 2(r'(x) - r(x))$, kde první nerovnost plyne z $r'(x) = r(z), r(x) \leq r(y)$, a druhá nerovnost plyne z konkávnosti logaritmu.
U zig-zigu se nám potenciál vrcholu x nezmění, takže cena je $1 + r'(y) + r'(z) - r(y) - r(z) \leq 2(r'(y) - r(y)) \leq 2(r'(y) - r(x))$. Tím pádem se to pak poteleskopuje na $2(r'(\text{nový kořen}) - r(x)) + 1$, a zbytek analýzy vyjde prakticky stejně.

Úloha 3 (Ultralíné splay stromy)
Motivováni úspěšnou leností, která vedla k operaci semisplay, chceme na stromě provádět ještě méně změn. Co kdybychom při splayování prvku x rotovali jen každou druhou hranu na cestě z x do kořene a postupně měnili splayovaný prvek?

Řešení
Tohle nevyjde, protože si můžeme postavit strom (přesněji cestu) tvaru "cikcak", kde vrcholy budou po řadě shora $1, n, 2, n-2, 3, n-3, \ldots$, a potom (při správné paritě n) splayování nejnižšího prvku strom jenom prohodí na strom tvaru $n, 1, n-1, 2, n-2, 3, \ldots$, takže se strom prakticky nijak nezmění.

Úloha 4 (Rozděl a spojuj)
Pro splay strom T a hodnotu k navrhněte operaci SPLIT, která strom T rozdělí na dva stromy T', T'', přičemž ve stromě T' jsou všechny hodnoty menší nebo rovny k a ve stromě T'' jsou všechny hodnoty větší než k. Pokuste se zachovat amortizovanou složitost.
Dále pro splay stromy T', T'', kde všechny hodnoty v T' jsou menší než všechny hodnoty v T'', navrhněte operaci JOIN(T', T''), která sloučí stromy do jednoho, opět při zachování amortizované složitosti.

Řešení a) SPLAY(k), resp LOOKUP(k), pak T' je kořen + levý podstrom, T'' je pravý podstrom.

b) Vysplayujeme maximum z T', a T'' přidáme jako pravého syna kořene.

Bonusové úlohy

Úloha 5 (Subset theorem)
Ukážeme následující tvrzení: mějme splay strom na vrcholech $[n]$ a podmnožinu $S \subseteq [n], s := |S|$. Pak m dotazů, které se ptají pouze na prvky S, má časovou složitost $\mathcal{O}(n \cdot s + m \log s)$.
Jako bonus jej můžeme zlepšit na $\mathcal{O}(n \cdot \min(s, \log n) + m \log s)$.
Neformálně důkaz postupuje následovně: rádi bychom si představili, že prvky S nám ve stromě tvoří malý podstrom (ideálně ve velikosti lineární v s), ve kterém se děje skoro vše důležité. Tak takový podstrom budeme sledovat s tím, že se tedy jedná o nejmenší podstrom, který obsahuje všechny prvky, na které jsme se dosud ptali. Ukážeme, že přidání jednoho nového vyhledaného prvku nám do podstromu přidá maximálně jeden další prvek, takže celý podstrom nebude příliš velký. Když už prvek v tom podstromě je, tak cenu všech dotazů můžeme počítat jenom v tom malém podstromě, ale když v něm ještě není, tak musíme cenu rozdělit na cenu mimo podstrom, a cenu v podstromě. Mimo podstrom můžeme cenu odhadnout triviálně. V podstromě pak můžeme použít větu o amortizované složitosti, a díky malé velikosti podstromu $\mathcal{O}(s)$ dostáváme celkový výsledek.
Předpokládejme, že máme posloupnost dotazů $q_1, \ldots, q_m$. Dále si označíme jako $T_0, \ldots, T_m$ stavy splay stromu, kde T_i je stav po i-té operaci, a jako $S_i := \{q_j : j \leq i\}, s_i := |S_i|$ si označíme množinu prvků, které byly dotazovány do i-té operace, a mohutnost této množiny. Budeme postupovat následovně:

a) Ukážeme, že když se podíváme na nejmenší podstrom T_i, který obsahuje všechny prvky S_i, tak tento podstrom obsahuje nejvýše $2s_i - 1$ vrcholů. Tento podstrom si dále označíme jako D_i. (D_0 je prázdný strom.)

b) Ukážeme, že to, co se děje při rotacích mezi T_i a T_{i+1} v D_i umíme simulovat v malém stromě, který odpovídá D_i.

c) Tím pádem můžeme cenu každého dotazu rozdělit na dvě části: cena na „zvednutí“ do D_i a cena uvnitř D_i. Triviálně odhadněte cenu na zvednutí s prvků jako $\mathcal{O}(n \cdot s)$.

d) Odhadneme za pomoci nevážené analýzy cenu všech operací uvnitř $D = (D_0, \ldots, D_m)$ (z prvního bodu máme odhad na velikost D_m) a uvědomíme si, že to stačí.

Bonus: Využijme věty ze cvičení a uvědomme si, že pro velké s ji také můžeme použít se stejným výsledkem.

Bonus 2: Jak to vypadá s částí a) pro naivní splay?

Řešení a) Indukcí podle počtu provedených operací, na začátku je strom prázdný, po první operaci má jenom jeden vrchol. Dále se podíváme na nějaký obecný dotaz. Pokud prvek už v podstromu je, tak se jedná o normální vysplayování uvnitř podstromu.

Jinak hledaný prvek v podstromu zatím není: pak se podíváme na první dvojrotaci, ve které se vyskytuje nějaký prvek podstromu (všechny předchozí na podstrom nemají žádný vliv): jediný případ, kdy můžeme přidat další vrchol, nastane při zig-zig, kde nejvyšší vrchol už v podstromě je a prostřední vrchol v podstromě ještě není. (Je to prostě o projití několika případů.) A když už v podstromě jsme, tak nemůžeme žádný vrchol přidat (ale můžeme potenciálně nějaké vrcholy odebrat, což nám horní odhad nezkazí).

b) Simulace: budeme si udržovat pouze ten malý podstrom, a když do něj budeme chtít přidat nový vrchol (protože jsme na něj právě poprvé provedli vyhledání), tak přidáme nejvýše dva vrcholy tak, aby všechny rotace odpovídaly rotacím ve velkém stromě. (Takže když provádím zig-zig, kde jenom nejvyšší vrchol už byl ve stromě, tak provedu „dvojitý insert“, kde přidám dva vrcholy najednou, což se ale stejně

uamortizuje.) Zároveň, pokud se mi stane, že se nějaký prvek, který ještě nebyl vyhledán, stane listem, tak ho z podstromu smažu (tohle čistě technicky dělat nemusím, ale usnadňuje to definici toho, že D_i je nejmenší podstrom; zároveň mazání jsou z pohledu potenciálu operace vlastně zdarma, takže mi to cenu nezmění).

c) Každý prvek je přinejhorším v hloubce n, takže celková cena je $\mathcal{O}(n \cdot s)$.

d) Začínáme s prázdným stromem, a končíme se stromem o s prvcích, takže cena z amortizované analýzy je maximálně $\mathcal{O}(s \log s + m \log s)$, a protože $m \geq s$, máme $\mathcal{O}(m \log s)$.

Bonus: Můžeme použít working set theorem, a potom si vybereme mez podle velikosti s.

Bonus 2: Strom se tam taky vytvoří, a dokonce bude mít velikost jenom s_i.