4. cvičení

Datové struktury I, 25. 10. 2022 https://iuuk.mff.cuni.cz/~chmel/2223/ds1/

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

1. Jak je definován potenciál splay stromu?
2. Jaká je amortizovaná cena rotace (dvojité a jednoduché)?
3. Jaká je amortizovaná cena celého splaye a jak plyne z amortizovaných cen rotace?
4. Jaké je *reálná* cena celého splaye (a v jakých jednotkách ji vlastně počítáme)?
5. Jaký je potenciál perfektně vyváženého stromu? (Stačí nám rozumný horní a dolní odhad, pro jednoduchost předpokládejte $n = 2^k - 1$.)
6. Jaký je potenciál cesty? (Opět stačí rozumné odhady.)

Řešení 1. 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.

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

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

4. Počet provedených rotací - v rotacích.

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

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

Úloha 2 (Hloubka splay stromů nemusí být vždy logaritmická)
Navrhněte posloupnost operací, která vytvoří splay strom s lineární hloubkou a to jak pro naivní splay, tak pro správný splay.

Řešení
Postupně pustíme operace Splay(1), Splay(2),...,Splay(n), nebo místo Splayů Findy.
Alternativně, můžeme takhle insertovat do prázdného stromu.

Úloha 3 (Naivita se ne vždy vyplácí)
Co se pokazí, když operaci Splay implementujeme naivně, tedy jen pomocí jednoduchých rotací jedné hrany?

Řešení
Pokud budeme dvakrát za sebou volat Splay(1),...,Splay(n), tak nejdříve vytvoříme levou cestu, a potom s jednoduchými operacemi budeme stavět druhou levou cestu nad první, takže celková cena bude $\mathcal{O}(n^2)$.

Úloha 4 (Rozdělujeme stromy)
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.

Řešení
Splay(k), pak T' je kořen + levý podstrom, T'' je pravý podstrom.

Úloha 5 (A teď stromy spojujeme)
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 v čase $\mathcal{O}(|T'| + |T''|)$.

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

Úloha 6 (Insert a delete přes split a join)
Navrhněte alternativní operace INSERT a DELETE za pomoci operací SPLIT a JOIN.

Řešení
INSERT: Vysplayujeme danou hodnotu k. Pokud tam byla, tak nic neděláme, jinak provedeme split na kořen, k bude náš nový kořen, T' bude levý podstrom a T'' bude pravý podstrom.
DELETE: Vysplayujeme danou hodnotu k. Pokud ve stromě nebyla, tak neděláme nic, jinak splitneme na k, k jako kořen odmažeme (protože ze SPLITu má jenom levého syna), a oba stromy zase přes JOIN spojíme.

Bonusové úlohy

Úloha 7 (Líně vyvažujeme)
Na přednášce jste viděli binární vyhledávací stromy vyvažované líně pomocí lokálního přestavování. Prozkoumáme detaily přestavby a navrhneme implementaci operace delete:

a) Ukažte, že libovolný BVS lze přestavět na perfektně vyvážený v lineárním čase.

b) Potenciál v amortizované analýze byl definován skrz součet potenciálu vrcholů a potenciál vrcholu $r(v)$ je buď $|s(L_v) - s(R_v)|$, nebo nula, pokud je předchozí rozdíl přesně jedna. Co by se rozbilo, kdybychom pro rozdíl jedna prostě nastavili potenciál na jedničku?

c) Přidejte delete tak, jak funguje v obyčejném BST spolu s vhodným lokálním přestavením a analyzujte složitost.

d) Přidejte delete pomocí náhrobků a globální přestavby stromu. To znamená, že mazané vrcholy necháme ve stromě, ale označíme je jako smazané (náhrobek). Je-li náhrobků ve stromě moc, celý strom přestavíme a přitom náhrobky zlikvidujeme. Doplňte detaily (např. kolik je „moc“) a konstrukci analyzujte.

Řešení a) V lineárním čase projdeme všechny prvky přes následníka, z toho si postavíme pole, a z toho potom postavíme vyvážený BVS v lineárním čase podle úlohy z prvního cvičení.

b) I rozdíl 1 může být perfektně vyvážený, takže bychom potom mohli stále mít lineární potenciál.

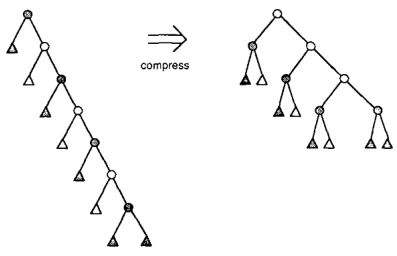
c) Vyjde to přibližně podobně: delete sníží velikostí všech podstromů na cestě do kořene o 1, takže potenciál se může zvýšit přinejhorším o 2. Pokud jsme si nic nerozbili, tak vše v pořádku, jinak provedeme rebuild, a ono to vyjde úplně stejně jako u insertu.

d) Vezmeme jako „moc“ $n/8$, jako potenciál budeme mít součet potenciálů vrcholů plus $8\times$ počet náhrobků. Už víme, že na přestavení potřebujeme lineární čas, ale zbavíme se tím všech náhrobků + ještě si můžeme snížit potenciály vrcholů, čímž se to zaplatí.

Úloha 8 (Pro fajnšmekry)
Proveďte přestavění BVS na perfektně vyvážený BVS v lineárním čase *a s konstantní pamětí.*

Řešení 1. ze stromu uděláme pravou cestu

2. vhodně pozkomprimujeme listy



Zdroj: `https://web.eecs.umich.edu/~qstout/pap/CACM86.pdf`