

Cvičení #7 BVS stromy

Miloš Chromý

chromy@ktiml.mff.cuni.cz

1. **Perfektní vyváženost.** Navrhněte algoritmus, který ze setříděného pole vyrobí v lineárním čase dokonale (vrcholově) vyvážený BVS v čase $O(n)$. (Počet vrcholů v levém podstromu se liší od počtu vrcholů v pravém podstromu maximálně o 1.)
2. **Fronta na poště.** Při modernizaci pošty stát vypsál výběrové řízení na informační systém obsluhy zákazníků. Systém musí zvládnout realizovat frontu pro právě jednu přepážku a měl by podporovat následující operace. Přidat nově příchozího zákazníka s novým číslem do fronty (číslo by mělo být nejvyšší z již udělených).
 - Odebrat zákazníka z fronty (buď si to rozmyslel a odešel, takže ve frontě již nečeká nebo přišel na řadu a je obslužen).
 - Přidat zákazníka se stávajícím číslem zpět do fronty (vrátil se ze záchoda, ale systém ho neprávem odstranil z fronty).

Zkuste si rozmyslet, jestli vám pomůže BVS a jak ho použít na strukturu (jak provádět jednotlivé operace). Jaká je časová složitost těchto operací? (Uvažujte nevyvážený BVS.)

3. **Další prosím.** Jak v takovém systému pro daného zákazníka najít dalšího, který čeká hned za ním?
4. **Výpis fronty.** Pošta zainvestovala a koupila displej, na kterém chce zobrazit všechna čísla zákazníků čekajících ve frontě. Navrhněte jak vypsát všechny čekající zákazníky v lineárním čase.
 - (a) Pomocí rekurze.
 - (b) Pomocí funkce následníka.
5. **Nová přepážka.** Na poště je velký nátlak a tak pošta otevře novou přepážku. Novou frontu utvoříme tak, že všechna čísla větší než s dáme do nové fronty a ostatní zůstanou ve frontě původní.

Navrhněte operaci Split, která dostane BVS T a hodnotu s , a rozdělí strom na dva BVS T_1 a T_2 takové, že hodnoty v T_1 jsou menší než s a hodnoty v T_2 jsou větší než s .
6. **Klátící korona.** Korona zasáhla i poštu a tak se musí jedna ze dvou přepážek zavřít a existující fronty sloužit do jedné. Zákazníci byli řazeni do fronty náhodně, a tudíž neplatí žádné pravidlo pro čísla zákazníku mezi dvěma frontami.

Navrhněte operaci Join, která dostane dva BVS T_1 , T_2 a sloučí jejich obsah do jediného BVS. Algoritmus by měl pracovat v čase $O(|T_1| + |T_2|)$.

7. **Zrychlování.** Pošta se v dobách karantény stala oblíbeným místem a lidé si chodí stoupnout do fronty pro pocit ze sociální sounáležitosti. Tím se nám mnohanásobně zvýšil počet zákazníků a jednotlivé operace trvají příliš dlouho. Navrhněte algoritmus, který v lineárním čase BVS reprezentující frontu dokonale vyváží. *Jak to udělat s konstantní pamětí?*
8. **Kolikátý ve frontě jsem.** Pro frontu na poště jsme zvolili AVL strom. Vzhledem k délce fronty by zákazníci rádi věděli, kolikátý ve frontě jsou, aby si mohli odskočit, dojít se vyspat, nebo vynalézt lék na koronu.
Upravte AVL stromy tak, aby dokázaly pro libovolné číslo zákazníka zjistit, kolik lidí je ve frontě před ním.
9. **Korupce za časů korony.** Někteří zákazníci nechtějí čekat a tak se rozhodnou uplácat, čímž se dostanou ze své současné pozice na pozici první (kde je opět někdo může předběhnout). Sestrojte datovou strukturu pro uložení seznamu tak, abychom uměli rychle najít k -tý prvek a přesunout ho na začátek. (AVL strom?)
10. **Maxima.** Mějme AVL strom použitý jako slovník: v každém vrcholu sídlí klíč a nějaká celočíselná hodnota. Upravte strom, aby uměl rychle zjistit největší hodnotu přiřazenou nějakému klíči z intervalu $[a, b]$.
11. **Maxima2.** Pokračujme v předchozím cvičení: Také chceme, aby strom uměl ve všech vrcholech s klíči v zadaném intervalu $[a, b]$ rychle zvýšit hodnoty o δ . Může se hodit princip líného vyhodnocování.