NTIN060: Algoritmy a Datové struktury **14.4.2020**

Cvičení #9: Hash brown

Miloš Chromý chromy@ktiml.mff.cuni.cz

1. **Join.** Navrhněte operaci `join(X, Y)`,která dostane dva (a, b)-stromy X a Y a sloučí je do jednoho. Může se přitom spolehnout na to, že všechny klíče z X jsou menší než všechny z Y . Zkuste dosáhnout složitosti $O(\log |X| + \log |Y|)$.

2. **Split.** Navrhněte operaci `split(`T`,`x`)`, která zadaný (a, b)-strom T rozdělí na dva stromy. V jednom budou klíče menší než x, v druhém ty větší. Pokuste se o logaritmickou časovou složitost.

3. **Prázdná paměť.** Nevýhodou (a, b)-stromů je, že plýtvají pamětí – může se stát, že vrcholy jsou zaplněné jen z poloviny. Navrhněte úpravu, která zaručí zaplnění z alespoň 2/3.

4. **Přičítání jedničky 2.** Přičtení 1 k binárnímu číslu N může mít až složitost $\log N$, vzhledem k přenosu. Dokažte, že $n\times$ přičtení jedničky bude mít amortizovanou složitost $O(1)$ na jednu operaci přičtení.

5. **Přičítání jedničky 3.** Dokažte, že přičítání 1 je amortizovaně konstantní i pro číslo v ternární soustavě.

6. **Nafukovací pole.** Pole, které má operaci `push`, která vezeme prvek, dá ho na konec a poku je pole plné, tak zdvojnásobí jeho kapacitu. Jaká je jeho amortizovaná složitost?

7. **Vynafukovací pole.** Pole u kterého můžu dělat nejen operaci `push`, ale i `pop`. Navrhněte implenmentaci s konstantní amortizovanou složitostí.

8. **Arnold Schwarzeneger.** Máme zásobník, kde funkce `push` i `pop` má konstantní časovou složitost. Přidáme operaci `popall` která zavolá tolikrát `pop`, kolik je právě prvků v zásobníku. Dokažte, že při provedení n operací $o_1, \ldots, o_n$, kde o_i je buď `push` nebo `popall`, je amortizovaná složitost každé operace konstantní, tedy $O(1)$.