

1 Intervalový update.

Mějme AVL strom jako slovník dvojic (klíč, hodnota), přičemž hodnoty jsou číselné. Upravte jej, aby podporoval operaci $\text{add}(x, y, \delta)$, která k hodnotám všech klíčů v intervalu $[x, y]$ přičte δ . Tato operace má běžet v $O(\log n)$, takže nemusíme hned provést aktualizaci hodnot všech klíčů v daném intervalu, stačí když $\text{Find}(k)$ vrátí správnou hodnotu klíče k .

2 (a, b) -stromy a volba b .

Odhalte, jak závisí složitost operací s (a, b) -stromy na parametrech a a b . Z toho odvoďte, že se nikdy nevyplatí volit b výrazně větší než $2a$.

3 (a, b) -join.

Navrhněte operaci $\text{Join}(X, Y)$, 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 $O(\log |X| + \log |Y|) = O(\max\{\log |X|, \log |Y|\})$.

4 Lineární konstrukce.

Ukažte, že pokud budeme do prázdného (a, b) -stromu postupně vkládat klíče $1, \dots, n$, provedeme celkem $\Theta(n)$ operací. K tomu si potřebujeme pamatovat, ve kterém vrcholu skončil předchozí vložený klíč, abychom nemuseli pokaždé hledat znovu od kořene.

5 Okénkový medián.

Máme dáno přirozené k a poté na vstupu přicházejí čísla. Kdykoliv přijde další, vypište medián z posledních k čísel (medián je $\lceil k/2 \rceil$ -tý nejmenší prvek). Dosáhněte časové složitosti $O(\log k)$ na operaci.

6 Jednosměrné operace.

Navrhněte úpravu operací Insert a Delete do (a, b) -stromů, aby operace procházely stromem pouze od kořene dolů a nebylo nutné jít zpět. Předpokládejte $b \geq 2a$.

7 Jen jeden bit pro AVL.

V AVL stromu je potřeba udržovat v každém vrcholu znaménko $\{-, 0, +\}$, na což jsou potřeba dva bity. Ukažte, že existuje způsob, jak si vystačit s jedním bitem na vrchol.

Bonusové úlohy:

8 Malé zaplnění.

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 $2/3$, až na zaokrouhlování a kromě kořene.

9 Pomalé sjednocení.

Dokažte, že budeme-li reprezentovat množiny binárními vyhledávacími stromy, nelze sjednocení provést rychleji než lineárně v nejhorším případě. Platí to dokonce i tehdy, máme-li na vstupu zaručený dokonale vyvážený strom a výstup může být jakkoliv nevyvážený.