ADS cvičení 9
(a,b) a červenočerné stromy
18. 4. 2018

1. Dokažte, že procházíme-li obecný vyhledávací strom v symetrickém pořadí vrcholů, pravidelně se střídají vnitřní vrcholy s vnějšími. To znamená, že obsahují-li vnitřní vrcholy klíče $x_1, \dots, x_n$, pak vnější vrcholy odpovídají intervalům $(-\infty, x_1), (x_1, x_2), (x_2, x_3), \dots, (x_n, +\infty)$.

2. 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. Ukažte, že pokud budeme do prázdného 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.

4. Spočítejte přesně, jaká může být minimální a maximální hloubka LLRB stromu s n klíči.

5. LLRB stromy jsou asymptoticky stejně rychlé jako AVL stromy. Zamyslete se nad jejich rozdíly při praktickém použití.

6. Uspořádejme všechny permutace na množině $\{1, \dots, n\}$ lexikograficky. Vymyslete algoritmus, který pro dané k sestrojí v pořadí k-tou permutaci v čase $O(nlogn)$. Navrhněte též převod permutace na její pořadové číslo.

7. Vymyslete jiné uspořádání všech permutací, v němž půjde mezi permutací a jejím pořadovým číslem převádět v lineárním čase.

8. 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 jakkoli nevyvážený.

9. Na vstupu postupně přicházejí čísla. Kdykoliv přijde další, vypište medián z posledních k čísel. Dosáhněte časové složitosti $O(logk)$ na operaci.

10. Sestrojte datovou strukturu pro uložení seznamu tak, abychom uměli rychle najít k-tý prvek a přesunout ho na začátek.

ADS cvičení 9
(a,b) a červenočerné stromy
18. 4. 2018

1. Dokažte, že procházíme-li obecný vyhledávací strom v symetrickém pořadí vrcholů, pravidelně se střídají vnitřní vrcholy s vnějšími. To znamená, že obsahují-li vnitřní vrcholy klíče $x_1, \ldots, x_n$, pak vnější vrcholy odpovídají intervalům $(-\infty, x_1), (x_1, x_2), (x_2, x_3), \ldots, (x_n, +\infty)$.

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

4. Spočítejte přesně, jaká může být minimální a maximální hloubka LLRB stromu s n klíči.

5. LLRB stromy jsou asymptoticky stejně rychlé jako AVL stromy. Zamyslete se nad jejich rozdíly při praktickém použití.

6. Uspořádejme všechny permutace na množině $\{1, \ldots, n\}$ lexikograficky. Vymyslete algoritmus, který pro dané k sestrojí v pořadí k-tou permutaci v čase $O(nlogn)$. Navrhněte též převod permutace na její pořadové číslo.

7. Vymyslete jiné uspořádání všech permutací, v němž půjde mezi permutací a jejím pořadovým číslem převádět v lineárním čase.

8. 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 jakkoli nevyvážený.

9. Na vstupu postupně přicházejí čísla. Kdykoliv přijde další, vypište medián z posledních k čísel. Dosáhněte časové složitosti $O(logk)$ na operaci.

10. Sestrojte datovou strukturu pro uložení seznamu tak, abychom uměli rychle najít k-tý prvek a přesunout ho na začátek.