

Líný AVL strom

Po velkém úspěchu se správou holých větví v BVS se tým profesora Fabinarise Suchbauma soustředí na studium speciálních variant AVL stromů. Jmenovitě zkoumají přednosti tzv. líného AVL stromu. Tento strom implementuje operaci Insert stejně jako standardní AVL strom, liší se ale v operaci Delete, která pouze označuje uzly jako smazané, aniž by měnila strukturu stromu. Pro zachování dobrých vlastností stromu je ale potřeba označené uzly čas od času fyzicky smazat, což se děje během tzv. konsolidační fáze, jež je spuštěna, pokud maximální počet označených uzlů na cestě z kořene do listu dosáhne mezní hranice.

Notace

Pro líný AVL strom T , nechť $depth(T)$ označuje hloubku T , která je definovaná jako počet hran na nejdelší cestě z kořene do některého z listů. Pokud je T prázdný, pak $depth(T) = -1$. Pro uzel X stromu T , nechť $key(X)$ označuje jeho celočíselný klíč. Předpokládáme, že každý uzel má booleovský příznak určující, zda klíč reprezentovaný v daném uzlu je smazaný, či ne. Nechť $deleted(X)$ označuje hodnotu tohoto příznaku pro uzel X . Říkáme, že X je *označený jako smazaný* právě tehdy, když $deleted(X)=true$.

Insert(k)

Pokud je klíč k uložený v T v uzlu X , potom operace Insert pouze nastaví $deleted(X):=false$. Když klíč k není v T , je do T vložen podle stejných pravidel jako v případě standardního AVL stromu.

Delete(k)

Pokud je klíč k uložený v T v uzlu X , potom operace Delete pouze nastaví $deleted(X):=true$. Když klíč k není v T , potom operace neprovede žádnou změnu.

Konsolidace

Konsolidace proběhne automaticky po operaci Insert nebo Delete, pokud tato operace způsobí, že T obsahuje cestu z kořene do listu, která má alespoň $1+\lfloor depth(T)/2 \rfloor$ uzlů označených jako smazaných. Konsolidace probíhá sekvečně v iteracích, dokud v T existuje uzel označený jako smazaný. V každé iteraci je vždy odstraněn uzel X označený jako smazaný, který je mezi takovými uzly aktuálně první v pořadí postorder. Odstranění uzlu se provede zavoláním operace $Remove(X)$ definované níže.

Remove(X)

Pro fyzické smazání uzlu X z líného AVL stromu T se aplikují rekurzivní pravidla:

- Když je X list, proved' standardní Delete v AVL stromu s případným vyvážením pomocí rotací. Pokud lze aplikovat jednoduchou i dvojitou rotaci, jednoduchá rotace má vždy přednost.
- Když má X levého potomka, pak v levém podstromu uzlu X najdi uzel Y s největším klíčem. Následně nastav $key(X):=key(Y)$, $deleted(X):=false$ a proved' $Remove(Y)$.
- Když nemá X levého potomka, ale má pravého, pak v pravém podstromu uzlu X najdi uzel Y s nejmenším klíčem. Následně nastav $key(X):=key(Y)$, $deleted(X):=false$ a proved' $Remove(Y)$.

Pro ilustraci průběhu operací uvažujme následující příklad líného AVL stromu:

```

      2+
     1+  3+
        4+
  
```

Strom obsahuje klíče 1, 2, 3, 4 a jeho hloubka je 2. Symbol "+" za klíčem indikuje, že příslušný uzel není označený jako smazaný. Když se provede $Delete(3)$, struktura stromu se nezmění. Obdržíme výsledek

```

      2+
     1+  3-
        4+
  
```

kde symbol "-" za klíčem 3 indikuje, že daný uzel obsahuje klíč označený jako smazaný. Operace $Insert(5)$ vloží nový uzel a provede jednoduchou rotaci potřebnou k vyvážení uzlu s klíčem 3.

```

      2+
     1+  4+
        3-  5+
  
```

Operace $Insert(3)$ nalezne uzel s klíčem 3 a nastaví jeho příznak $deleted$ na $false$.

```

      2+
     1+  4+
        3+  5+
  
```

Provedení $Delete(5)$ a $Delete(2)$ vytvoří strom

```

      2-
    1+   4+
      3+   5-

```

což vede k aktivaci konsolidační fáze, neboť nyní strom obsahuje cestu z kořene do listu, která má $2 \geq 1 + \lfloor 2/2 \rfloor$ uzlů označených jako smazaných. Příslušná konsolidační procedura smaže nejprve uzel s klíčem 5. Výsledkem je strom

```

      2-
    1+   4+
      3+

```

Poté je smazán kořen. Klíč v kořeni je nahrazen klíčem 1 z levého podstromu a uzel původně obsahující tento klíč je odstraněn.

```

    1+   4+
      3+

```

Následuje použití dvojité rotace v kořeni k získání výsledného vyváženého stromu.

```

      3+
    1+   4+

```

Úloha

Implementujte líný AVL strom. Nad iniciálně prázdným líným AVL stromem proveďte zadanou posloupnost operací Insert a Delete. Spočítejte, kolik bylo během celého procesu provedeno rotací a konsolidací, zjistěte také hloubku výsledného stromu.

Vstup

První vstupní řádek obsahuje kladné celé číslo N , které určuje celkový počet operací Insert a Delete, jež se mají vykonat. Každý z následujících N řádků obsahuje právě jedno nenulové celé číslo. Pokud je to kladné číslo K , potom daný řádek reprezentuje operaci *Insert*(K). V případě záporného čísla K reprezentuje řádek operaci *Delete*($|K|$). Operace se mají provést ve stejném pořadí, v jakém se na vstupu vyskytují.

Platí $1 \leq N \leq 2 \times 10^6$ a $1 \leq |K| \leq 1.5 \times 10^6$.

Výstup

Výstup sestává z jednoho řádku obsahujícího tři celá čísla D , R , C oddělená mezerami. Číslo D je hloubka výsledného stromu. Číslo R je odvozené od celkového počtu rotací vykonaných během celého procesu (včetně rotací provedených při konsolidacích). Každá jednoduchá rotace přispívá do R hodnotou 1, každá dvojitá rotace přispívá hodnotou 2 (tzn., dvojitou rotaci započítáváme jako dvě jednoduché rotace). A konečně, číslo C odpovídá počtu vyvolaných konsolidací.

Příklady

Vstup	Vstup	Vstup
9	10	12
2	5	2
1	8	4
3	7	1
4	10	5
-3	4	8
5	6	7
3	-8	-4
-5	-10	-7
-2	-5	-2
	-6	-5
		-1
		-8
Výstup	Výstup	Výstup
1 3 1	1 3 2	-1 2 4

Veřejná data

Veřejná data k úloze jsou k dispozici. Veřejná data jsou uložena také v odevzdávacím systému a při každém odevzdání/spuštění úlohy dostává řešitel kompletní výstup na stdout a stderr ze svého programu pro každý soubor