

Intervalový binární vyhledávací strom

Profesor Fabinaris Suchbaum z institutu Maxe Plancka v Saarbrückenu je uznávaným odborníkem na binární vyhledávací stromy a jejich modifikace. V nedávné době se zabýval otázkou, jak pomocí stromových struktur reprezentovat množinu intervalů. Výsledkem jeho výzkumu je návrh takzvaného intervalového binárního vyhledávacího stromu (iBVS). Jedná se o binární strom, který reprezentuje množinu celočíselných, po dvou navzájem disjunktních intervalů následovně:

- Každému uzlu U jsou jako klíč přiřazena dvě celá čísla a, b , kde $a \leq b$. Tato čísla reprezentují celočíselný interval $[a, b] = \{a, a+1, \dots, b\}$.
- Pokud má uzel U levého potomka s klíčem $[a_L, b_L]$, potom platí $b_L < a$. Analogicky, pokud má U pravého potomka s klíčem $[a_R, b_R]$, platí $b < a_R$. Tímto je zajištěno, že klíče ve stromě skutečně odpovídají po dvou navzájem disjunktním intervalům.

Pro iBVS jsou definovány operace Insert a Delete. Insert(a, b) vkládá do stromu T klíč $I=[a, b]$ podle těchto pravidel:

- Pokud je I disjunktní se všemi klíči v T , pak je přidán nový uzel (list) s klíčem I stejným způsobem jako v případě BVS.
- Necht' I_1, \dots, I_k jsou všechny klíče v T , se kterými má I neprázdný průnik. Označme $J = I \cup I_1 \cup \dots \cup I_k$. Pro všechna $i=1, \dots, k$, označme uzel s klíčem I_i jako V_i . Bez újmy na obecnosti předpokládejme, že V_1 má mezi uzly V_i nejmenší hloubku.

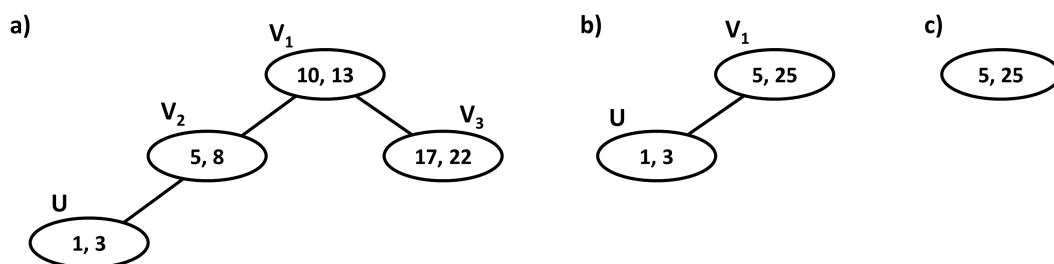
Výsledkem vložení I do T je iBVS T^* , který vznikne z T po těchto modifikacích:

- Strom T^* je tvořen všemi uzly T kromě uzlů V_2, \dots, V_k , které se odstraní (označme $S = \{V_2, \dots, V_k\}$).
- Uzel V_1 má v T^* klíč J .
- Pokud má uzel $U \notin S$ v T rodiče, který je v S , pak jeho novým rodičem v T^* je nejbližší uzel $P \notin S$ na cestě v T z U do kořene. Je-li v T uzel U v levém podstromu uzlu P , stane se v T^* levým potomkem P , je-li v pravém podstromu P , stane se pravým potomkem.

Delete(c) maže ve stromě uzel s klíčem $[a, b]$, který splňuje $a \leq c \leq b$. Pokud takovýto uzel neexistuje, neprovede operace Delete žádnou modifikaci. Pokud existuje, je uzel odstraněn podle pravidel BVS, s tím, že při mazání uzlu U se dvěma potomky se náhradní klíč hledá jako minimum v pravém podstromu uzlu U .

Úloha

Nad původně prázdným iBVS proveďte danou posloupnost operací Insert a Delete. Zjistěte základní údaje o výsledném stromu.



Obrázek 1. a) Stav původně prázdného iBVS po provedení operací Insert(10,13), Insert(5,8), Insert(17,22), Insert(1,3). b) Stav po následném provedení Insert(7,25). Ve stromě zůstaly uzly označené jako V_1 a U . Uzly V_2 a V_3 se odstranily, uzel U se stal levým potomkem uzlu V_1 , jehož klíč se změnil na $[5,25] = [7,25] \cup [5,8] \cup [10,13] \cup [17,22]$. c) Stav po následném provedení Delete(2).

Vstup

První řádek vstupu obsahuje celé číslo N , které udává počet provedených operací. Následuje N řádků, které reprezentují operace, přičemž je zachováno pořadí jejich provedení. Řádek reprezentující operaci Delete(k) obsahuje postupně znak 'd', mezeru a číslo k . Řádek reprezentující operaci Insert(k_1, k_2) obsahuje postupně znak 'i', mezeru, číslo k_1 , mezeru a číslo k_2 .

Platí $N \leq 1.5 \times 10^6$. Parametry operací Delete a Insert jsou nezáporná celá čísla menší než 2.1×10^6 .

Výstup

Výstup obsahuje jeden textový řádek, na kterém jsou čísla U , H , M oddělená mezerou. U je počet uzlů výsledného stromu, H je jeho hloubka a M je počet celých čísel reprezentovaných stromem. (Pozn.: Pokud uzel má klíč $[a, b]$, pak reprezentuje celkem $b - a + 1$ čísel.)

Příklad 1

Vstup

```
6
i 10 13
i 5 8
i 17 22
i 1 3
i 7 25
d 2
```

Výstup

```
1 0 21
```

Data a řešení Příkladu 1 můžeme vidět na **Obrázku 1**.

Příklad 2

Vstup

```
5
i 2 2
i 1 1
i 4 5
d 2
i 3 6
```

Výstup

```
2 1 5
```

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 veřejných dat.

[Veřejná data](#)