

DualAVL vyhledávací strom

Profesor Fabianis Suchbaum vede již dlouhou dobu mezinárodně úspěšný tým zkoumající vlastnosti binárních vyhledávacích stromů a jejich netriviálních modifikací. Tým působí v ústavu Max Planck Institute for Software Systems, v Saarbrücken v Německu. Dřívější významné příspěvky týmu si lze prohlédnout [zde](#) a [zde](#) a [zde](#) a [zde](#) a [zde](#) a [zde](#). V aktuálním grantu zkoumá tým variantu AVL stromu nazvanou dualAVL. V této variantě mohou uzly stromu obsahovat více klíčů, čímž se snižuje hloubka stromu, která je kritická pro rychlé vyhledávání a úpravy ve stromu.

Základní vlastnosti dualAVL stromu jsou:

1. Pokud strom obsahuje 1 až 3 klíče, pak jsou všechny klíče uloženy v kořeni, který je zároveň jediným uzlem ve stromu.
2. Když strom obsahuje více než 3 klíče, pak každý vnitřní uzel obsahuje právě dva klíče, zatímco každý list může obsahovat 1, 2 nebo 3 klíče.
3. Po každé změně počtu uzlů v dualAVL stromu se kontroluje jeho vyvážení. Pravidla pro vyvážení a aplikaci rotací jsou identická se standardním AVL stromem.

Jsou definovány dvě jednoduché operace dělení listu a kontrakce uzlu.

Dělení listu L probíhá vždy, když L obsahuje 3 klíče a má do něj být vložen další klíč k . Označme M množinu klíčů L sjednocenou s $\{k\}$. Při dělení L se vytvoří dva nové uzly $L1$ a $L2$, přičemž $L1$ se stane levým potomkem L a $L2$ se stane pravým potomkem L . Klíč s hodnotou $\min(M)$ se vloží do $L1$, klíč s hodnotou $\max(M)$ se vloží do $L2$ a zbylé dva klíče v M se vloží do L místo jeho původního obsahu.

Kontrakce uzlu U probíhá vždy, když U je vnitřní uzel a oba jeho bezprostřední potomky jsou listy, které oba obsahují jediný klíč a jeden z těchto klíčů má být smazán. Z uzlu U se stane list obsahující původní klíče U a ten z klíčů v obou potomcích, který nemá být smazán. Oba potomky U se odstraní ze stromu i s jejich obsahem.

Pro libovolný uzel U stromu T označme

$\min(U)$, $\max(U)$ klíč s minimální a s maximální hodnotou v U ,

$U.L$ a $U.R$ a $U.P$ levého a pravého bezprostředního potomka a rodiče uzlu U ,

$\text{height}(U)$ výšku podstromu T s kořenem U . (Výška je maximální vzdálenost z U do některého z listů v podstromu s kořenem U .)

Dále označme kořen stromu T symbolem $T.\text{root}$.

Operace **Insert**(k) ve stromu T je totožná s operací $\text{Insert}(k, T.\text{root})$.

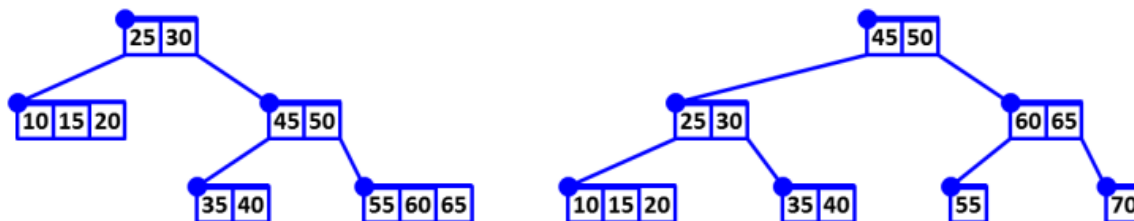
Operace $\text{Insert}(k, U)$, kde U je uzel T , probíhá podle pravidel:

- 1. Pokud k je obsažen ve stromu, operace nemá žádný efekt.
- 2. Pokud strom obsahuje 0, 1 nebo 2 klíče, $\text{Insert}(k)$ vloží k do kořene T , který je zároveň jediným uzlem T .
- 3. Pokud je U vnitřní uzel T :
 - 3a. Pokud $k < \min(U)$, provede se $\text{Insert}(k, U.L)$
 - 3b. Pokud $\max(U) < k$, provede se $\text{Insert}(k, U.R)$
 - 3c. Pokud $\min(U) < k < \max(U)$, pak v případě $\text{height}(U.L) \leq \text{height}(U.R)$ se provede $\text{Insert}(\min(U), U.L)$ a v U se klíč $\min(U)$ nahradí klíčem k , jinak se provede $\text{Insert}(\max(U), U.R)$ a v uzlu U se klíč $\max(U)$ nahradí klíčem k .
- 4. Pokud je U listem s jedním nebo dvěma klíči, k se přidá do U .
- 5. Pokud je U listem se třemi klíči provede se dělení listu U .

Operace **Delete**(k) ve stromu T je totožná s operací $\text{Delete}(k, T.\text{root})$.

Operace $\text{Delete}(k, U)$, kde U je uzel T , probíhá podle pravidel:

- 1. Pokud k není obsažen ve stromu, operace nemá žádný efekt.
- 2. Pokud strom obsahuje 1, 2, nebo 3 klíče, $\text{Delete}(k)$ odstraní k z kořene T , který je zároveň jediným uzlem T .
- 3. Pokud je U vnitřní uzel T a U neobsahuje k , provede se buď $\text{delete}(k, U.L)$ nebo $\text{delete}(k, U.R)$ podle toho, zda $k < \min(U)$ nebo $\max(U) < k$.
- 4. Pokud je U vnitřní uzel T a U obsahuje k , klíč k se z U odstraní a nahradí se U klíčem y , který leží v některém listu Ly . Pokud platí $\text{height}(U.L) \leq \text{height}(U.R)$, je y klíč s minimální hodnotou v $U.R$, jinak je y klíč s maximální hodnotou v $U.L$. Poté se provede $\text{delete}(y, Ly)$.
- 5. Pokud je U listem se dvěma nebo třemi klíči, klíč k se odstraní z U .
- 6. Pokud je U listem s jedním klíčem a jeho sourozenec je vnitřní uzel, označme z ten klíč v $U.P$, jehož hodnota je nejbližší ke k . Klíč k se v U nahradí klíčem z a provede se $\text{delete}(z, U.P)$.
- 7. Pokud je U listem s jedním klíčem a jeho sourozenec V je také list se dvěma nebo třemi klíči, označme M množinu klíčů v $U.P$ a ve V . Klíč k se odstraní z U a klíče množiny M se distribuují do uzlů $U, V, U.P$ tak, aby U obsahoval jediný klíč.
- 8. Pokud je U listem s jedním klíčem a jeho sourozenec V je také list s jedním klíčem, provede se kontrakce uzlu $U.P$.
- 9. Při vyvažování stromu pro operaci delete má vždy, pokud je to možné, přednost jednoduchá rotace před rotací dvojitou.



Obrázek 1. Ukázka dualAVL stromu. Po vložení klíče 70 do stromu vlevo vznikne strom vpravo. Klíč 70 nelze umístit do listu s klíči 55, 60, 65, proto se nejprve provede dělení tohoto listu. Tím se hloubka pravého podstromu kořene zvýší o 1, a proto je poté aplikována levá rotace v kořeni. Ukázka je součástí Příkladu 1. níže.

Úloha

Simulujte operace Insert a Delete v dualAVL stromu.

Vstup

První řádek obsahuje jedno celé číslo C určující počet příkazů. Každý z následujících C řádků obsahuje jeden příkaz. Příkaz má tvar $I\ x\ d\ m$ nebo $D\ x\ d\ m$, kde x, d, m jsou celá čísla a znaky I a D jsou velká písmena anglické abecedy. Příkazy postupně mění obsah jediného dualAVL stromu, který je před provedením prvního příkazu prázdný.

Příkaz $I\ x\ d\ m$ znamená, že budou provedeny postupně operace $\text{Insert}(x + k \cdot d)$, pro $k = 0, 1, 2, \dots, \lfloor (m - x) / d \rfloor$. (Výraz $\lfloor w \rfloor$ představuje dolní celou část reálného čísla w .) Například, příkaz $I\ 3\ 6\ 30$ určuje postupné provedení operací $\text{Insert}(3)$, $\text{Insert}(9)$, $\text{Insert}(15)$, $\text{Insert}(21)$, $\text{Insert}(27)$.

Analogicky, příkaz $D\ x\ d\ m$, kde x, d, m jsou celá čísla, znamená, že budou provedeny postupně operace $\text{Delete}(x + k \cdot d)$, pro $k = 0, 1, 2, \dots, \lfloor (m - x) / d \rfloor$.

Platí $1 \leq C \leq 10^4$. Pro každý příkaz platí $1 \leq x \leq m \leq 10^6$, $1 \leq d \leq 20$.

Celkový počet operací Insert a Delete specifikovaných příkazy na vstupu nepřekročí $3 \cdot 10^7$.

Výstup

Výstup obsahuje jeden řádek se čtyřmi celými čísly oddělenými mezerou a představujícími, v tomto pořadí, počet uzlů, počet listů s jedním klíčem, počet listů se dvěma klíči a počet listů se třemi klíči ve výsledném dualAVL stromu po provedení všech operací specifikovaných na vstupu.

Example 1

Vstup

```
2
I 10 10 60
I 15 10 80
```

Výstup

```
7 2 1 1
```

```

      25,30
    -----
 10,15,20    40,50,60

```

INSERT (35)

```

      25,30
    -----
 10,15,20    40,50
              35    60

```

INSERT (45)

```

      25,30
    -----
 10,15,20    45,50
              35,40    60

```

INSERT (55)

```

      25,30
    -----
 10,15,20    45,50
              35,40    55,60

```

INSERT (65)

```

      25,30
    -----
 10,15,20    45,50
              35,40    55,60,65

```

INSERT (75)

```

      45,50
    -----
      25,30    60,65
    -----
 10,15,20    35,40    55    75

```

Schéma 1. DualAVL strom v Příkladu 1 v závěrečné fázi vývoje.

Kompletní průběh operací v Příkladu 1. a také v příkladu 2. a 3. je k dispozici v souboru veřejných dat níže.

Příklad 2

Vstup

```
3
I 10 2 30
I 50 2 70
D 20 2 60
```

Výstup

```
5 1 1 1
```

Příklad 3

Vstup

```
4
I 20 6 60
I 29 3 50
D 35 1 40
I 17 7 38
```

Výstup

```
5 0 0 3
```

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](#)