
Pembahasan Lab 5

Deksripsi Singkat

Diberikan sebuah list of permen yang memiliki harga dan tipe yang bisa di tambahkan atau kurangkan. Ada query BELI yang mencari permen dengan harga minimal dan maksimal diantara [L, R] dimana tipenya berbed

Ide Subtask 1 (TC 1 - 24)

Yang perlu disiapkan: AVL dimana setiap node menyimpan key (harga) dan frekuensi dari key tersebut. Karena semua tipenya berbeda, kita cukup mencari **lowerbound** dan **upperbound** harga permen di range L dan R.

Lowerbound: harga terkecil yang lebih besar atau sama dengan L

Upperbound: harga terbesar yang lebih kecil atau sama dengan R

- Jika lowerbound != upperbound, maka inilah jawaban dari querynya (cetak "lowerbound upperbound")
- Jika lowerbound == upperbound, maka cek frekuensi node tersebut. Jika frekuensi = 1, maka jawabannya -1 -1. Jika frekuensi > 1, maka jawabannya "lowerbound upperbound". Karena ada dua permen dengan tipe beda, meskipun harganya sama

Pencarian lowerbound dan upperbound dapat dilakukan dalam $O(\log N)$ pada AVL tree

Ide Subtask 2 (TC 25 - 32)

Selain membuat AVL tree harga semua tipe, dibuat juga AVL tree harga untuk masing-masing tipe. Karena ada 8 tipe beda, maksimal ada tambahan 8 AVL beda.

Pertama kita cari lowerbound dan upperbound di AVL tree utama

Ada 2 case yang perlu diperhatikan:

- Jika tipe permen di lowerbound != upperbound, maka outputkan "lowerbound upperbound"
- Jika tipenya sama, maka optimalnya ada dua kemungkinan:
 - Mengambil permen lowerbound, dan permen harga tertinggi yang tipenya beda
 - Mengambil permen upperbound, dan permen harga terendah yang tipenya beda

Untuk mencari nilai-nilai tersebut, kita bisa melakukan query lowerbound/upperbound pada setiap AVL tipe selain tipe yang sama tadi. Kompleksitasnya per query adalah $O(8 * \lg N)$

Ide Subtask 3 (TC 33 - 40)

Kita membuat setup AVL sama seperti subtask 2 (AVL utama, dan AVL per tipe). Selain itu, setiap AVL juga menyimpan ukuran (banyak key) dari dia dan subtreenya.

Namun jika melakukan metode subtask 2 akan mendapatkan TLE karena tipenya bisa banyak. Sehingga kita harus menggunakan observasi berikut:

- Misalkan `avl.orderOfKey(X)` menyatakan banyaknya key pada avl tree yang nilainya $\leq X$. Ini bisa dicari dalam $O(\lg N)$ dengan memanfaatkan nilai "size" pada avl tree.
- Misalkan `avl.inRange(L, R)` adalah sebuah fungsi yang menghitung banyaknya permen dengan harga diantara $[L, R]$ di avl tree "avl". Ini dapat dihitung dengan `avl.orderOfKey(R) - avl.orderOfKey(L-1)`

Saat BELI, ada dua case sama seperti subtask 2, namun cara handle case keduanya beda. Katakan T sebagai tipe permen dari lowerbound dan upperbound.

Misalkan `bigAVL` = avl tree semua harga, `smallAVL` = avl tree dari tipe T. Maka, untuk suatu range $[L, R]$, kita bisa cek apakah ada permen dengan tipe bukan T dengan cek apakah `bigAVL(L, R) - smallAVL(L, R) > 0`. Dengan properti ini, kita bisa melakukan **binary search** untuk mencari nilai lowerbound dan upperbound. Kompleksitas untuk setiap BELI adalah $O(\log^2 N)$, dengan overall $O(Q \log^2 N)$:

```
bigAVL = # avl tree besar
smallAVL = # avl tree tipe T

# cek apakah ada value berbeda di [L, R]
D = bigAVL.inRange(L, R) - smallAVL.inRange(L, R)
If D == 0:
    Return "-1 -1"

# mencari upper bound
Left = L+1, right = R-1
While left < right:
    Mid = (left + right)/2s

    D = bigAVL.inRange(mid, R) - smallAVL.inRange(mid, R)
    If D > 0:
        Left = mid
    Else:
        Right = mid-1
Upperbound = Left # harga terbesar yang memiliki permen dengan tipe selain T

# mencari lower bound mirip
```

