FAKULTAS ILMU KOMPUTER

CSGE602040 - Struktur Data dan Algoritma
Semester Ganjil - 2023/2024
Lab 7
Deadline: Kamis, 23 November 2023, 17.00 WIB

Gunting Batu Kertas

Deskripsi

Kairi, salah satu mahasiswa Universitas Indonesia, kini sedang bosan karena Internet di kostnya sedang *down*. Namun, ia mengingat bahwa ia memiliki teman yang sedang berada di kost yang sama dengannya. Oleh karena itu, Kairi memutuskan untuk mengajak temannya bermain suatu permainan yang bernama Gunting Batu Kertas, di mana permainan tersebut memanfaatkan *box* yang memiliki id*, value,* dan *state*. Id bersifat unik yang sesuai dengan urutan *box* ditambahkan (dimulai dari 0), *value* merupakan nilai dari *box* tersebut, sedangkan *state* merupakan salah satu di antara ROCK, PAPER, atau SCISSOR.

Dalam permainan Gunting Batu Kertas ini, terdapat tiga perintah yang dapat dilakukan, yaitu sebagai berikut:

1. **A V S**
   Pemain dapat menambahkan *box* dengan *value* V dan *state* S ke dalam permainan.
2. **D I J**
   Pemain dapat memilih b_I untuk berkompetisi dengan b_J. Kemenangan akan didasarkan pada *state* dari *box* yang berkompetisi. Berikut adalah cara menentukan *box* yang menang:
   - *Box* dengan *state* ROCK akan menang melawan *box* dengan *state* SCISSOR
   - *Box* dengan *state* PAPER akan menang melawan *box* dengan *state* ROCK
   - *Box* dengan *state* SCISSOR akan menang melawan *box* dengan *state* PAPER

   Jika b_I menang melawan b_J, maka V_I akan **bertambah** sebanyak V_J, dan V_J akan **dibagi 2 (pembulatan ke bawah)** dari nilainya sekarang. Hal tersebut berlaku juga sebaliknya. Jika **id I sama dengan id J** atau ***state box* yang berkompetisi sama**, maka tidak ada perubahan yang terjadi pada *box* tersebut.
3. **N I**
   Pemain dapat memilih b_I untuk berkompetisi dengan box tetangganya. Jika b_I menang melawan b_{I-1}, V_I akan **bertambah** sejumlah V_{I-1}, sementara V_{I-1} tetap. Jika kalah, kedua *value* tersebut **tidak berubah**. Aturan yang sama berlaku saat berkompetisi dengan b_{I+1}. Jika tidak ada box tetangga, tidak ada kompetisi. Jika ***state box* yang berkompetisi sama**, maka tidak ada perubahan yang terjadi pada *box* tersebut.

Setiap perintah yang dilakukan akan menampilkan ***value*** dan ***state*** dari *box* dengan ***value* tertinggi** yang terdapat dalam permainan. Jika terdapat lebih dari satu *box* dengan *value* yang sama dan merupakan *value* tertinggi, maka yang akan ditampilkan adalah *box* dengan **id terkecil**.

Penjelasan Terkait Singkatan

- b_N : *Box* dengan id N
- V_N : *Value box* dengan id N

Format Masukan

- Baris pertama berisi bilangan bulat N yang menyatakan banyaknya *box* di awal yang terdapat dalam permainan.
- N baris berikutnya berisi pasangan V_I S_I yang merupakan *value* dan *state* untuk *box* dengan id I.
- Baris berikutnya berisi bilangan bulat T yang menyatakan banyaknya perintah yang akan dilakukan.
- T baris berikutnya berisi perintah yang telah dijelaskan di atas.

Format Keluaran

T baris, dimana baris ke-i merupakan pasangan *value* dan *state* dari *box* dengan *value* tertinggi yang terdapat dalam permainan setelah melakukan perintah ke-i.

Batasan

- $1 \leq N \leq 10^4$
- $1 \leq T \leq 10^6$
- $-2 \times 10^9 \leq V \leq 2 \times 10^9$, di mana $V \in \mathbb{Z}$
- $S \in$ {“R”, “P”, “S”}
- b_I dan b_J dijamin pasti ada

Contoh Masukan 1

```
3
10 R
5 S
15 P
3
A 7 S
D 0 2
N 1
```

Contoh Keluaran 1

```
15 P
25 P
30 S
```

Penjelasan

Di awal permainan, terdapat 3 *box*:

- Box dengan id 0 memiliki *value* 10 dan *state* ROCK
- Box dengan id 1 memiliki *value* 5 dan *state* SCISSOR
- Box dengan id 2 memiliki *value* 15 dan *state* PAPER

Kemudian, ada 3 perintah yang akan dilakukan:

- Tambah *box* dengan *value* 7 dan *state* SCISSOR

- *Box* dengan id 0 (ROCK) akan berkompetisi dengan *box* dengan id 2 (PAPER). PAPER menang melawan ROCK, maka value dari *box* dengan id 2 akan bertambah sebanyak value dari *box* dengan id 0 dan value dari *box* dengan id 0 akan dibagi 2.
- *Box* dengan id 1 (SCISSOR) akan berkompetisi dengan *box* di sebelahnya, yaitu *box* dengan id 0 (ROCK) dan *box* dengan id 2 (PAPER). SCISSOR kalah melawan ROCK, maka *value* dari *box* dengan id 1 dan *box* dengan id 0 tidak berubah. Kemudian, SCISSOR menang melawan PAPER, maka *value* dari *box* dengan id 1 akan bertambah sebanyak *value* dari *box* dengan id 2, sedangkan *value* dari *box* dengan id 2 tidak berubah.

Penjelasan hasil:

- Setelah perintah pertama, *box* dengan *value* tertinggi saat ini masih merupakan *box* dengan id 2 yang memiliki *value* 15 dan *state* PAPER, sehingga tercetak 15 PAPER.
- Setelah perintah kedua, *value* dari *box* dengan id 0 menjadi 5 (10 dibagi 2) dan *value* dari *box* dengan id 2 menjadi 25 (15+10). Oleh karena itu, *box* dengan *value* tertinggi saat ini adalah *box* dengan id 2 yang memiliki *value* 25 dan *state* PAPER, sehingga tercetak 25 PAPER.
- Setelah perintah ketiga, *value* dari *box* dengan id 1 menjadi 30 (5+25) karena SCISSOR menang melawan PAPER. Oleh karena itu, *box* dengan *value* tertinggi adalah *box* dengan id 1 yang memiliki *value* 30 dan *state* SCISSOR, sehingga tercetak 30 SCISSOR.

Contoh Masukan 2

```
3
10 S
9 R
10 P
6
A -5 S
A 10 R
D 1 3
A 14 S
N 2
A 14 R
```

Contoh Keluaran 2

```
10 S
10 S
10 S
14 S
14 P
14 P
```

Keterangan Tambahan

- Tidak boleh menggunakan struktur data selain yang telah diajarkan di kelas.
- Tidak boleh menggunakan *library* binary tree bawaan Java, seperti PriorityQueue<>(), TreeMap<>(), atau TreeSet<>().

Informasi Tambahan *Test Case*

Pada 45% *test case* berlaku:

- $1 \leq N \leq 10^2$
- $1 \leq T \leq 10^5$

Pada 55% *test case* berlaku:

- $10^2 < N \leq 10^4$
- $10^5 < T \leq 10^6$