

Perkembangan Warung Ikan Magetan

Riwayat Revisi:

- **Revisi 1** (07-11-2024 19.45): Penambahan keterangan constraint tambahan terkait struktur data dan algoritma sorting.
- **Revisi 2** (09-11-2024 09.40): Revisi penjelasan contoh masukan 2 dan penambahan penjelasan output untuk beberapa query



(Generated by Gemini AI: <https://gemini.google.com/app/>)

Deskripsi

Setelah kesuksesan Sofita dalam menjalankan bisnis “*Warung Ikan Magetan*”, Sofita berniat untuk memperluas jangkauan bisnisnya melalui *franchise* ke seluruh pelosok negeri. Namun kali ini Sofita ingin mengadakan sebuah kompetisi tim dimana tim yang menang akan mendapatkan hak *franchise* untuk membuka cabang “*Warung Ikan Magetan*”.

Dalam kompetisi tersebut, **peserta akan memiliki ID unik** dan Poin. Peserta-peserta kemudian akan dimasukkan ke dalam suatu tim yang minimal berisikan tujuh orang (Tim dengan jumlah kurang dari tujuh orang akan dieliminasi) dimana setiap **tim memiliki ID yang unik** dan **poin berdasarkan akumulasi poin peserta yang ada di tim**. Didalamnya peserta akan diurutkan secara *hierarchical*/menurun berdasarkan poin yang mereka miliki. Semakin besar poin peserta, maka tingkatan peserta tersebut akan semakin tinggi. Jika terdapat peserta dengan poin yang sama, maka peserta akan diurutkan berdasarkan jumlah pertandingan paling sedikit, jika peserta memiliki poin dan jumlah pertandingan yang sama maka selanjutnya akan diurutkan berdasarkan Id terkecil dari peserta tersebut. Selain itu Sofita juga menerapkan peraturan dimana peserta yang memiliki poin nol akan dieliminasi dari timnya.

Pada kompetisi tersebut, Sofita pada awalnya **akan mengawasi tim yang pertama kali terbentuk**. Sofita berhak berkeliling dari satu tim ke tim lain. Hal tersebut disebabkan karena adanya penjoki yang pada awalnya terdeteksi di tim yang memiliki poin paling kecil terus berkeliling selama kompetisi tersebut berjalan, sehingga Sofita sebagai pengawas bertugas untuk berkeliling agar menemukan tim yang menggunakan penjoki tersebut. Apabila Sofita menemukan penjoki pada suatu tim maka terdapat tiga konsekuensi,

1. Ketahuan 1x: Tiga peserta dengan poin terbesar dikeluarkan dari tim
2. Ketahuan 2x: Seluruh peserta pada tim diubah poinnya menjadi satu
3. Ketahuan 3x: Tim tersebut dieliminasi

Setelah itu, Sofita akan melempar penjoki keluar, namun sang penjoki tidak putus asa menawarkan jasanya dan masuk kembali ke tim yang memiliki poin terendah, apabila Sofita berada di tim dengan poin terendah, maka penjoki akan masuk ke tim dengan poin terendah kedua.

Dalam kompetisi terdapat berbagai kegiatan sebagai berikut.

1. A [JUMLAH_PESERTA]

Peserta pada tim yang sedang diawasi oleh sofita bertambah sebanyak **<JUMLAH_PESERTA>** dengan masing-masing peserta memiliki poin awal 3.

2. B [EXTREME_BOUND]

Sofita yang sekarang menginjak semester 3, saat ini sedang belajar mata kuliah statistika. Pada mata kuliah tersebut, dia sangat tertarik dengan materi *descriptive statistics*, khususnya pada konsep *Outlier*. Sofita pun ingin menghubungkan ilmu yang telah dipelajarinya dengan materi yang juga sedang dipelajarinya pada mata kuliah SDA. Keluaran akan bergantung pada nilai dari **<EXTREME_BOUND>**:

- Jika **EXTREME_BOUND = U** : Cari banyaknya peserta pada tim yang sedang diawasi sofita dengan nilai $> U$
- Jika **EXTREME_BOUND = L** : Cari banyaknya peserta pada tim yang sedang diawasi sofita dengan nilai $< L$
- Penjelasan lebih lanjut tentang variabel U dan L ada dibawah.

Sebagai klarifikasi, berikut cara menghitung nilai L dan U pada kumpulan bilangan:

- Urutkan kumpulan bilangan tersebut ke dalam suatu array. Sebut array tersebut adalah $A[0..(K - 1)]$ (Note: Perhatikan bahwa A terurut menaik, dan ukuran A adalah K .)
- Dari Array A , dapat dicari beberapa nilai:
 - $IndexQ1 = \max(0, \lfloor \frac{1}{4}K - 1 \rfloor)$
 - $IndexQ3 = \min(K - 1, \lfloor \frac{3}{4}K - 1 \rfloor)$ (Note: $\lfloor \rfloor$ adalah fungsi [floor](#))
 - $Q_1 = A_{IndexQ1}$
 - $Q_3 = A_{IndexQ3}$
 - $IQR = Q_3 - Q_1$
 - $L = Q_1 - \lfloor 1.5 * IQR \rfloor$
 - $U = Q_3 + \lfloor 1.5 * IQR \rfloor$

Perhatikan bahwa terdapat sedikit penyesuaian dalam penentuan beberapa variabel.

3. M [L/R]

Sofita berpindah tim kesebelah kanan atau kiri

- **M L** : Sofita berpindah ke tim sebelah kiri dari tim yang dia awas.
- **M R** : Sofita berpindah ke tim sebelah kanan dari tim yang dia awas.

Note: Apabila Sofita berada pada tim pertama maka **Move Left** akan memindahkan sofita ke tim terakhir. Selain itu, apabila sofita berada pada tim terakhir maka **Move Right** akan memindahkan sofita ke tim pertama.

4. T [ID_PESERTA_PENGIRIM] [ID_PESERTA_PENERIMA] [JUMLAH_POIN]

Pada tim yang sedang diawas sofita, peserta dengan **<ID_PESERTA_PENGIRIM>** mengirim poin sebanyak **<JUMLAH_POIN>** ke peserta dengan **<ID_PESERTA_PENERIMA>**. Dengan ketentuan jumlah poin maksimal yang dapat dikirim **lebih kecil** dari jumlah poin yang dimiliki pengirim serta ID penerima harus berada dalam 1 tim yang sama.

5. G [L/R]

Akan dibentuk sebuah tim baru yang beranggotakan tujuh orang dengan poin awal satu pada posisi **Kiri (L)** atau **Kanan (R)** dari tim yang sedang diawasi sofita saat ini.

6. V [ID_PESERTA1] [ID_PESERTA2] [ID_TIM] [HASIL]

<ID_PESERTA1> yang terdapat pada tim yang sedang diawas sofita bertanding dengan **<ID_PESERTA2>** yang terdapat di tim dengan ID **<ID_TIM>** memiliki **<HASIL>** pertandingan diantaranya,

- **0**: Seri, kedua peserta mendapat satu poin
- **1**: Peserta 1 menang, peserta 1 mendapat tiga poin, peserta 2 berkurang tiga poin

- -1: Peserta 2 menang, peserta 2 mendapat tiga poin, peserta 1 berkurang tiga poin

7. E [POIN]

Untuk lebih mudah dalam menentukan pemenang, Sofita akan mengeliminasi tim yang memiliki poin tim kurang dari **<POIN>**. Ketika tim yang sedang diawasi oleh sofita juga ikut terkena eliminasi, maka sofita akan mengawas tim dengan skor terbesar. Keluaran yang dihasilkan adalah **Jumlah tim yang dieliminasi**.

8. U

Sofita akan mencari jumlah peserta dengan poin yang unik pada tim yang sedang ia awasi. Keluaran yang dihasilkan adalah **Jumlah poin unik pada tim yang sedang diawasi**.

9. R

Karena tim memiliki poin yang terus berubah, maka Sofita sebagai pengawas tim akan melakukan pengurutan ulang tim tersebut berdasarkan Jumlah poin terbanyak. Jika terdapat tim memiliki poin yang sama maka akan diurutkan berdasarkan jumlah anggota tim paling sedikit. Jika masih terjadi kesamaan, urutkan berdasarkan ID tim terkecil. Ketika sudah dilakukan pengurutan posisi dari penjoki akan mengikuti tim terakhir yang sedang ia tempati. Sedangkan posisi sofita akan kembali ke tim dengan peringkat paling tinggi. Keluaran yang dihasilkan adalah **id tim yang sedang diawasi Sofita**.

Note: Tim dengan urutan **terbesar** akan berada di sebelah **paling kiri**.

10. J [DIRECTION]

Penjoki akan berpindah menawarkan jasanya dari 1 tim ke tim yang berada di sampingnya. Penjoki akan berpindah mengikut **<DIRECTION>** dimana

- Jika **DIRECTION = L** : maka Penjoki akan berpindah ke samping kiri dari tim yang ditempati sekarang.
- Jika **DIRECTION = R** : maka Penjoki akan berpindah ke samping kanan dari tim yang ditempati sekarang.

Terdapat kondisi sebagai berikut.

- Penjoki dapat mengetahui keberadaan sofita, sehingga ketika tempat tujuan penjoki ada Sofita yang sedang mengawas maka Penjoki akan menetap di tim tersebut.

Keluaran yang dihasilkan adalah **ID tim dari tempat penjoki berada**.

Note: Apabila Penjoki berada pada tim pertama maka **Move Left** akan memindahkan penjokike tim terakhir. Selain itu, apabila penjoki berada pada tim terakhir maka **Move Right** akan memindahkan penjoki ke tim pertama.

Format Masukan

- Baris pertama berisi M , yaitu banyaknya tim yang ikut berpartisipasi dalam kompetisi.
 - Setiap tim akan memiliki id. Id tim akan di-generate ketika sebuah tim didaftarkan ke kompetisi dengan *increment* 1 dari id tim terakhir. Id Tim akan diawali dengan 1.
- Baris kedua adalah M bilangan M_i yang dipisahkan dengan spasi
 - M_i merupakan jumlah peserta dalam tim ke- i .
 - Setiap peserta akan memiliki sebuah id. Id peserta akan di-generate ketika seorang peserta dimasukkan ke dalam tim dengan cara *increment* 1 dari id peserta terakhir yang masuk ke dalam tim. Id peserta dimulai dari 1.
- Baris ketiga berisi $\sum M_i$ bilangan P_j yang dipisahkan dengan spasi.
 - P_j merupakan poin awal dari siswa dengan id j .
- Baris berikutnya berisi sebuah bilangan Q , banyaknya perintah yang dilakukan.
- Q baris selanjutnya berisi aktivitas yang sesuai dengan format aktivitas yang telah dijabarkan sebelumnya.

Format Keluaran

- Keluaran dari perintah A
 - Cetak **jumlah peserta** pada tim setelah ditambahkan peserta baru
 - Cetak **-1** apabila sofita sedang tidak mengawasi tim manapun
- Keluaran dari perintah B
 - Cetak **jumlah peserta** yang memenuhi syarat yang ditentukan Sofita
- Keluaran dari perintah M
 - Cetak **ID tim** yang diawasi Sofita setelah pindah
 - Cetak **-1** apabila Sofita tidak sedang mengawasi tim manapun
- Keluaran dari perintah T
 - Cetak **poin** peserta pengirim dan peserta penerima (dipisah dengan *whitespace*)
 - Cetak **-1** apabila salah satu atau kedua ID Peserta tidak ditemukan
 - Cetak **-1** apabila jumlah poin yang dikirim < jumlah poin pengirim
- Keluaran dari perintah G
 - Cetak **ID tim** yang baru saja berpartisipasi
- Keluaran dari perintah V
 - Cetak **poin peserta** yang menang
 - Jika seri, Cetak **poin dari kedua peserta** (dipisah dengan *whitespace*)
 - Cetak **-1** apabila salah satu atau kedua ID Peserta tidak ditemukan
- Keluaran dari perintah E
 - Cetak **jumlah tim** yang terkena eliminasi.
- Keluaran dari perintah U
 - Cetak **jumlah peserta** yang memiliki nilai unik dalam tim yang diawasi sofita.
 - Cetak **-1** apabila sofita sedang tidak mengawasi tim manapun
- Keluaran dari perintah R
 - Cetak **ID tim** dari tempat Sofita mengawas.
 - Cetak **-1** apabila sofita sedang tidak mengawasi tim manapun

- Keluaran dari perintah J
 - Cetak **ID tim** dari tempat penjoki memberikan layanannya.
 - Cetak **-1** apabila Penjoki sedang tidak membantu tim manapun

Batasan

- $2 \leq M \leq 10^5$
- $7 \leq M_i \leq 10^5$
- $14 \leq \Sigma M_i \leq 6 \times 10^5$
- $0 \leq P_j \leq 1000$
- $1 \leq Q \leq 2 \times 10^5$

Batasan Kueri Aktivitas

- $\langle \text{BANYAK_QUERY_E} \rangle \leq 50$
- $\langle \text{BANYAK_QUERY_R} \rangle \leq 50$

Contoh Masukan 1

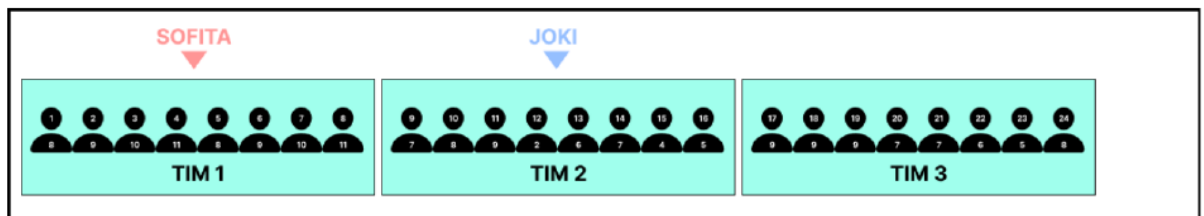
```
3
8 8 8
8 9 10 11 8 9 10 11 7 8 9 2 6 7 4 5 9 9 9 7 7 6 5 8
8
A 1
M L
J R
U
M L
G L
M L
E 20
```

Contoh Keluaran 1

```
9
3
2
5
1
4
4
1
```

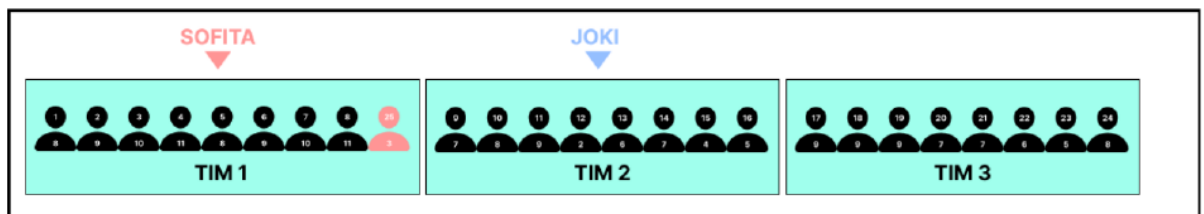
Penjelasan Contoh 1

1. Kondisi Awal



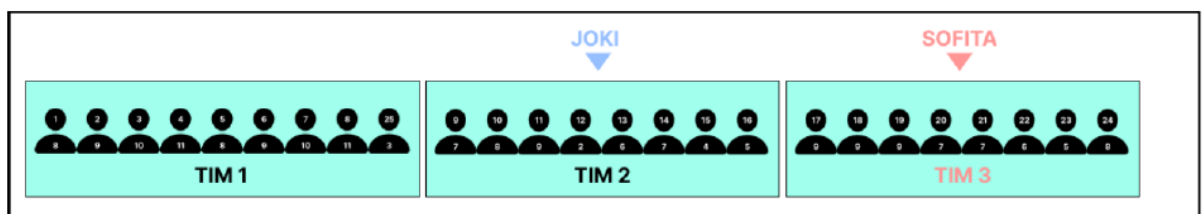
Terdapat 3 tim dengan masing-masing jumlah 8, 8, 8 dengan jumlah masing-masing poin sesuai dengan poin yang diinput. Sofita akan selalu berada pada tim nomor 1, sementara Joki akan berada pada tim dengan poin paling rendah. Pada kasus ini, poin tim 1 adalah 79, tim 2 adalah 48, dan tim 3 adalah 60, maka Joki akan berada pada tim 2

2. A 1



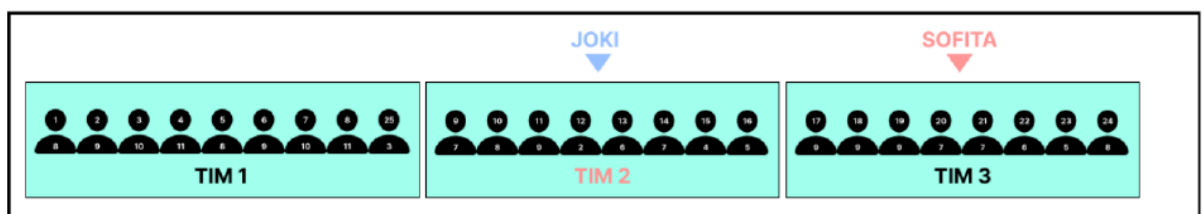
Terdapat peserta pada tim yang sedang diawasi oleh Sofita, cetak jumlah peserta pada tim yang sedang diawasi Sofita yaitu 9. Peserta dengan ID 25 akan terinisiasi dengan poin 3.

3. M L



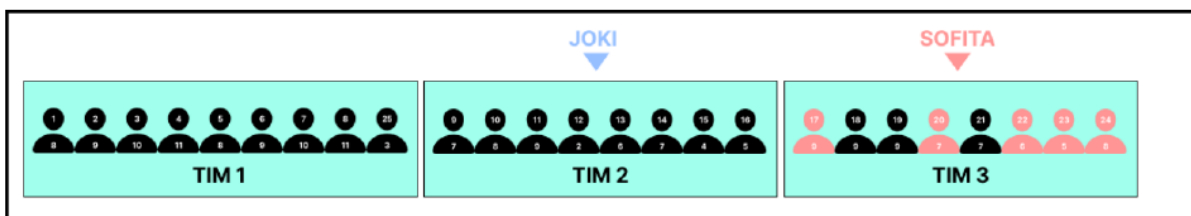
Sofita melakukan perpindahan ke kiri, maka cetak ID timnya, 3.

4. J R



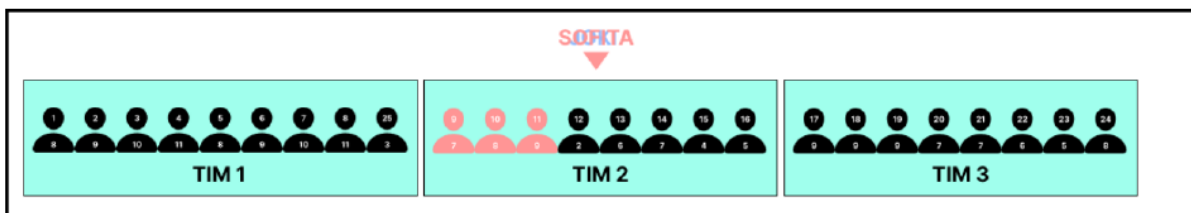
Tidak ada perubahan pada posisi Joki, karena ia tahu di tim kanannya sedang ada Sofita. Cetak ID tim posisi tetap Joki, 2.

5. U

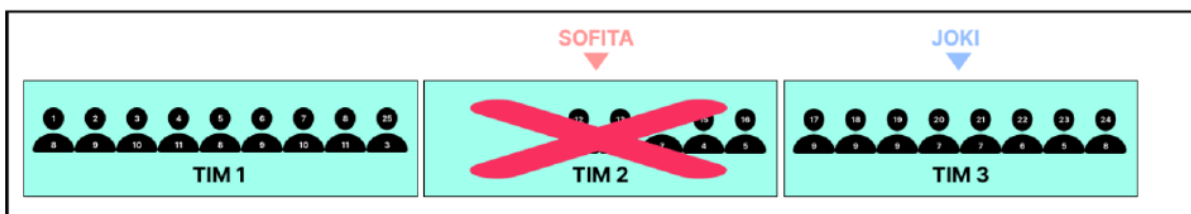


Pada tim yang sedang diawasi Sofita, terdapat 5 poin unik yaitu 9, 7, 6, 5, 8. Maka cetak 5.

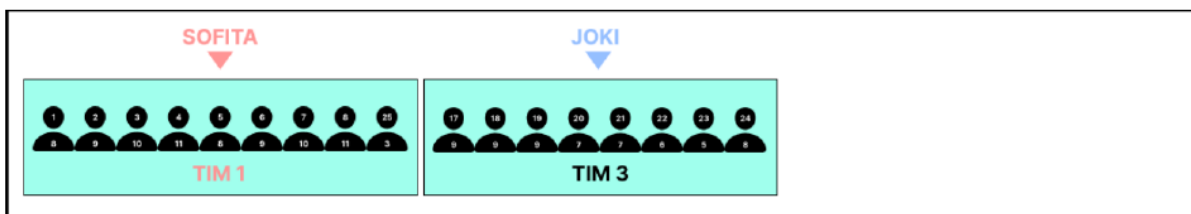
6. M L



Terjadi pertemuan pertama Joki dan Sofita pada Tim 2, tiga peserta dengan poin terbesar pada Tim 2 akan bersiap untuk dikeluarkan.

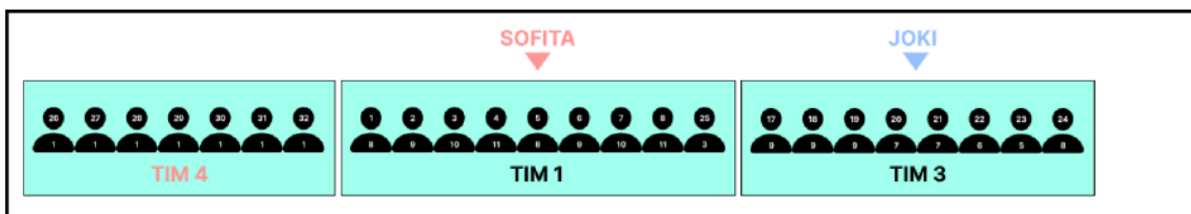


Terlihat bahwa Joki diusir oleh Sofita dan pergi. Namun karena setelah penghapusan, jumlah peserta pada Tim 2 hanya 5, maka tim tersebut akan dihapus, dan Sofita pergi pada kelas dengan total poin paling tinggi.



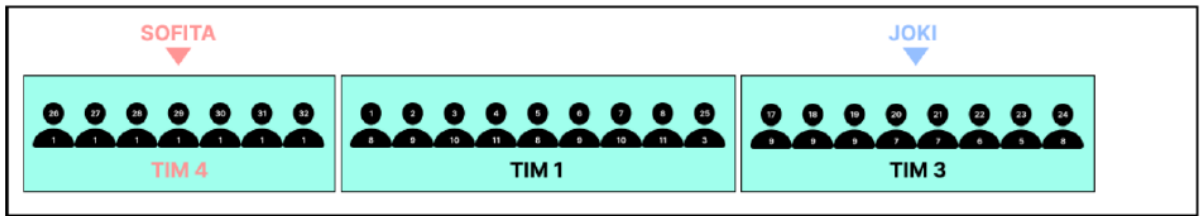
Posisi Sofita kembali dengan tim dengan poin paling tinggi. Cetak ID tim yaitu 1.

7. G L



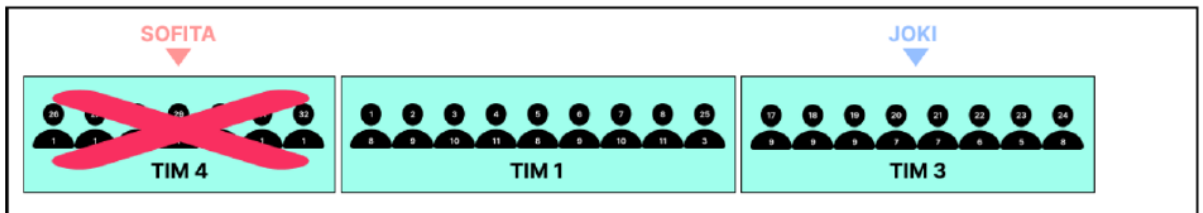
Muncul tim baru dengan ID 4, dengan 7 peserta. Masing-masing peserta memiliki poin 1, cetak ID tim, 4.

8. M L

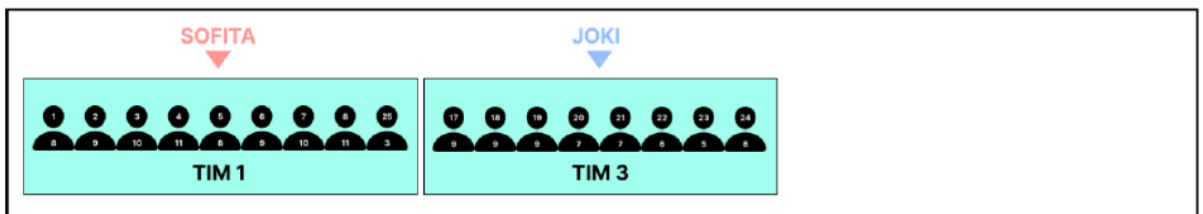


Sofita melakukan perpindahan ke kiri, maka cetak ID timnya, 4.

9. E 20



Poin tim 4 kurang dari 20. Oleh karena itu, tim tersebut di eliminasi. Cetak jumlah tim tereliminasi, 1.



Hasil akhir penghapusan tim.

Contoh Masukan 2

```

3
8 10 8
4 3 2 3 2 1 2 4 7 8 9 8 9 22 7 10 10 10 2 5 6 8 9 10 30 5
7
J L
T 21 3 2
V 1 4 1 1
R
R
A 3
B U

```

Contoh Keluaran 2

```

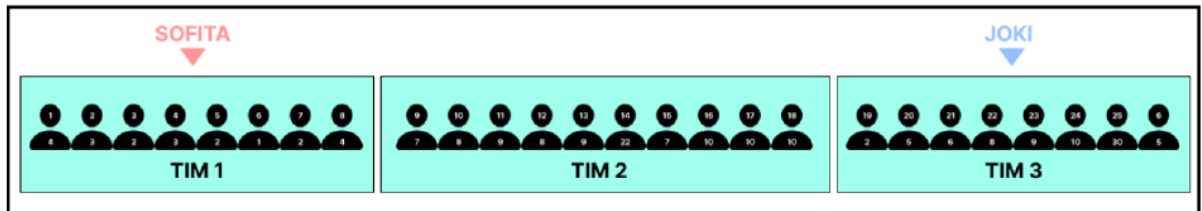
2
-1
7

```

2
3
11
1

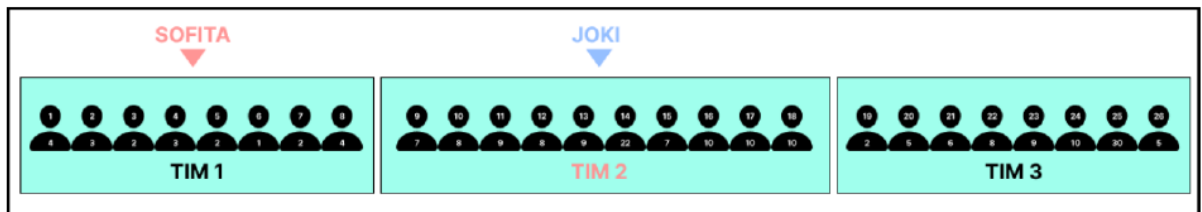
Penjelasan Contoh 2

1. Kondisi Awal



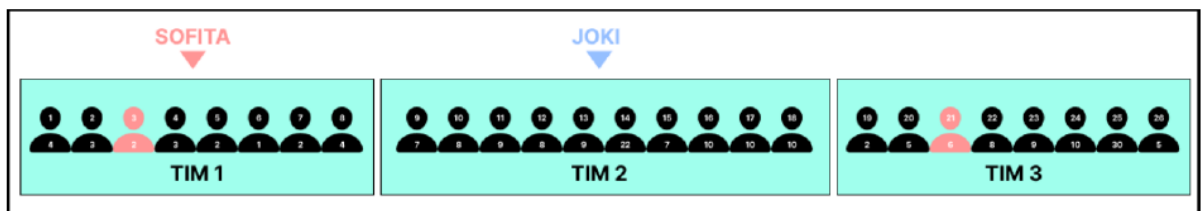
Terdapat 3 tim dengan masing-masing jumlah 8, 10, 8 dengan jumlah masing-masing poin sesuai dengan poin yang diinput. Sofita berada pada tim nomor 1, sementara Joki akan berada pada tim dengan poin paling rendah. Pada kasus ini, poin tim 1 adalah 21, tim 2 adalah 100, dan tim 3 adalah 75. Seharusnya, Joki berada pada posisi tim 1 yang memiliki poin paling rendah. Karena Joki tahu bahwa pada tim tersebut sedang diawasi oleh Sofita. Maka ia pindah ke tim 3 sebagai tim dengan poin terendah kedua.

2. JL



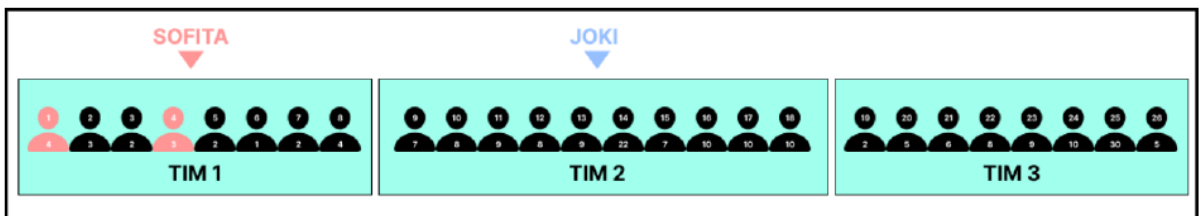
Joki melakukan perpindahan ke kiri. Cetak ID timnya, 2.

3. T 21 3 2

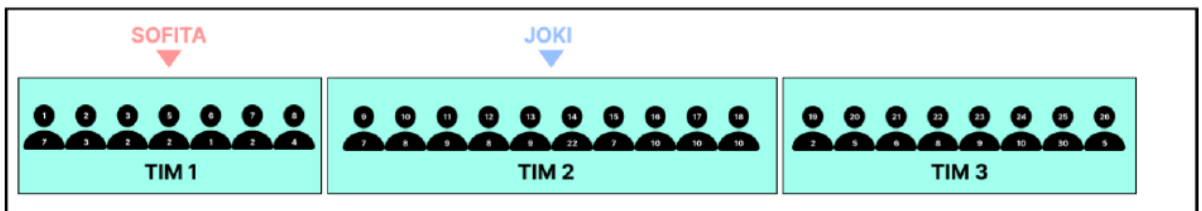


Dilakukan kueri untuk pemindahan poin dengan ID pengirim 21 dan penerima 2. Hal tersebut memang dapat dilakukan mengingat poin dari ID pengirim cukup untuk diberikan. Namun, perlu diingat bahwa sofita tidak berada pada tim 3. Sehingga cetak -1.

4. V 1 4 1

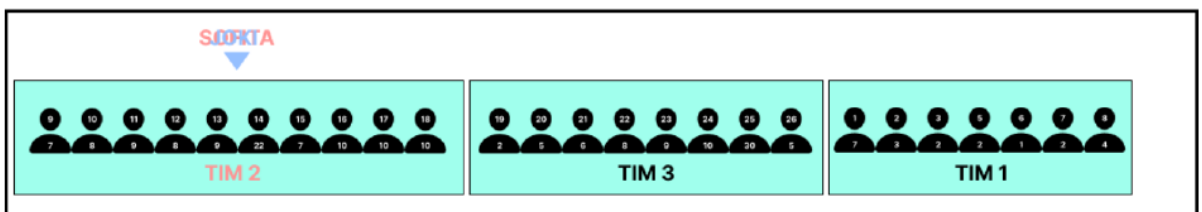


Dilakukan pertandingan dengan hasil menang untuk peserta 1. Artinya, peserta dengan ID 1 mendapatkan 3 poin, sehingga ponnya menjadi 7. Maka cetak 7. Sementara peserta 2 kehilangan 3 poinnya. Maka peserta dengan ID 4 poinnya 0. Oh tidak! Ia akan segera dieliminasi.

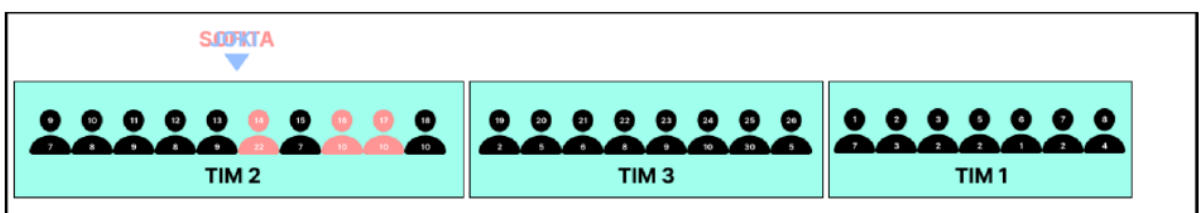


Hasil akhir setelah dilakukan kueri. Tim 1 berisi 7 orang karena adanya eliminasi pada peserta dengan ID 4.

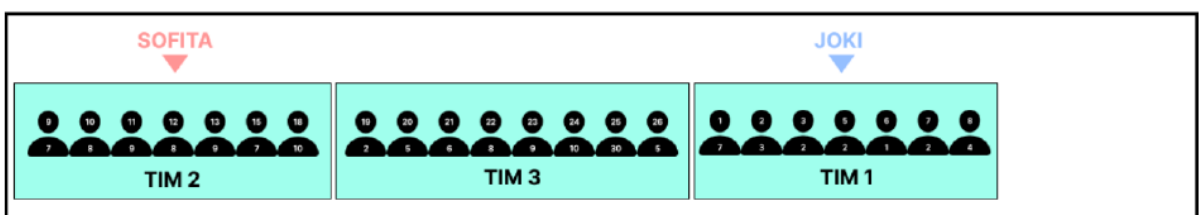
5. R



Terjadi pengurutan tim berdasarkan tim dengan poin tertinggi berada pada sebelah kiri, hal ini juga termasuk perpindahan Sofita. Dan terjadi lagi~~ pertemuan Sofita dengan Joki. Perlu dilakukan pengeluaran 3 peserta tertinggi pada tim 2. Cetak ID posisi Sofita sekarang, 2.

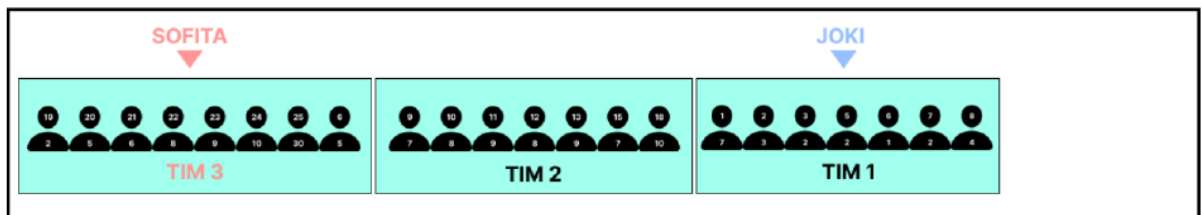


Pada tim 2, poin dengan nilai 22, 10, dan 10 dikeluarkan dalam tim, sehingga jumlah peserta pada tim 2 tersisa 7. Hal tersebut masih valid untuk jumlah peserta satu tim.



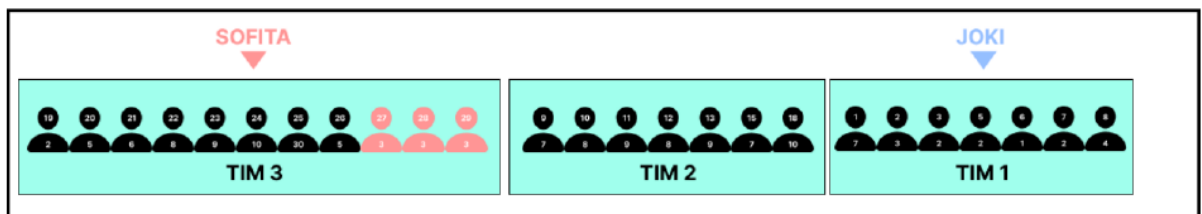
Hasil setelah kueri R dilakukan, posisi Joki akan pindah pada tim dengan poin terendah. Saat ini, poin tim 3 adalah 75, poin tim 2 adalah 58, sementara poin tim 1 adalah 21. Maka posisi akhir joki berada pada tim 1.

6. R



Terjadi lagi pengurutan tim. Saat ini, poin tim 1 adalah 21, poin tim 2 adalah 58, sementara poin tim 3 adalah 75. Maka posisi akhir Sofita berada pada tim dengan poin tertinggi yaitu tim 3. Cetak ID tim, 3.

7. A 3



Terdapat penambahan 3 peserta pada tim 3 dengan masing-masing inisialisasi poin 3. Cetak banyak jumlah peserta pada tim 3, 11.

8. B U

Pada kueri ini, dicari batas atas. Perhitungan akan sebagai berikut,

Data terurut = 2, 3, 3, 3, 5, 5, 6, 8, 9, 10, 30

diketahui bahwa Q1 dari data adalah 3 dan Q3 adalah 8. Maka didapatkan IQR sebesar 5. Batas atas dari tim ini adalah

$$U = Q_3 + [1.5 * IQR] = 8 + [1.5 * 5] = 15$$

Banyaknya peserta pada tim yang sedang Sofita awasi dengan poin > 15 adalah 1.

Informasi Tambahan Test-case

Testcase	Query
1-4	A
5-10	A, G
12-30	A, B, G, R
31-35	A, M, G, J
36-45	A, T, M, G, V
46-50	A, T, M, G, V, E
51-60	A, T, M, G, V, E, J
61-70	A, T, M, G, V, E, U, J

71-80	A, T, M, G, V, E, U, R, J
81-100	A, B, T, M, G, V, E, U, R, J

Keterangan Tambahan

Sesuai dengan scope materi TP 2, solusi yang Anda gunakan wajib mengimplementasikan:

- **Sorting**

Anda tidak diperbolehkan menggunakan bantuan library sorting java seperti Collections.sort() maupun menggunakan library PriorityQueue maupun TreeSet untuk menggantikan algoritma sorting pada query R. Anda wajib mengimplementasikan algoritma sorting (seperti Bubble Sort, Insertion Sort, algoritma sorting lainnya, maupun algoritma sorting buatan Anda sendiri) from scratch dan menggunakannya sebagai approach utama pada solusi query R.

- **LinkedList**

Anda wajib untuk mengimplementasikan LinkedList sendiri dan menggunakannya sebagai approach utama pada solusi. Tidak diperbolehkan menggunakan library LinkedList bawaan Java seperti LinkedList<>(), LinkedListDeque<>(), atau LinkedHashMap<>().

- **Tree (BST atau AVL)**

Anda wajib untuk mengimplementasikan Tree (BST atau AVL) sendiri dan menggunakannya sebagai approach utama pada solusi. Anda tidak diperbolehkan menggunakan struktur data ataupun algoritma tingkat lanjut yang belum atau tidak diajarkan di SDA, termasuk namun tidak terbatas pada: Segment Tree, Fenwick Tree, dan Square Root Decomposition.

Note: Konsekuensi terhadap pelanggaran keterangan tambahan akan dikenakan sanksi poin akhir 0