

C. Tur Menyenangkan

Time limit	2 s
Memory limit	512 MB

Deskripsi

Terdapat N buah atraksi di taman hiburan terbesar di Jakarta yang dinomori dari 0 hingga $N - 1$. Atraksi-atraksi ini dihubungkan oleh $N - 1$ jalan dua arah, sehingga terdapat suatu jalur unik untuk setiap pasang atraksi melalui jalan-jalan tersebut. Jalan-jalan tersebut dinomori dari 0 hingga $N - 2$. Jalan ke- i menghubungkan atraksi ke- $A[i]$ dan atraksi ke- $B[i]$, serta membutuhkan waktu satu jam untuk melaluinya. Untuk menghindari keramaian, setiap atraksi menjadi ujung dari paling banyak tiga jalan.

Anda ingin membuat sebuah tur yang mengunjungi setiap atraksi tepat sekali. Melalui banyak jalan saat berpindah dari satu atraksi ke atraksi lain tentu membosankan. Untuk membuat tur yang menyenangkan, Anda harus menemukan urutan semua atraksi, sehingga waktu yang diperlukan untuk mengunjungi atraksi selanjutnya tidak lebih lama dari waktu yang diperlukan untuk mengunjungi atraksi sebelumnya. Dengan kata lain, Anda ingin menemukan suatu barisan $P[0], P[1], \dots, P[N - 1]$ yang mengandung semua bilangan bulat dari 0 hingga $N - 1$ tepat sekali, sehingga waktu yang diperlukan untuk berpindah dari atraksi ke- $P[i]$ menuju atraksi ke- $P[i + 1]$ tidak lebih lama dari waktu yang diperlukan untuk berpindah dari atraksi ke- $P[i - 1]$ menuju atraksi ke- $P[i]$, untuk setiap $0 < i < N - 1$.

Anda tidak memiliki peta dari atraksi tersebut secara keseluruhan. Oleh karena itu, Anda harus menanyakan beberapa pertanyaan pada pusat informasi untuk membuat sebuah tur yang menyenangkan. Anda dapat bertanya paling banyak Q buah pertanyaan, dengan dua parameter X dan Y , dimana $0 \leq X, Y < N$. Setiap pertanyaan dapat berupa salah satu dari berikut ini:

- Berapa jam yang diperlukan untuk berpindah dari atraksi ke- X menuju atraksi ke- Y ? Tentunya, bila $X = Y$, jawabannya ialah 0.
- Berapa banyak atraksi Z , sehingga Anda harus mengunjungi atraksi ke- Y saat Anda berpindah dari atraksi ke- X menuju atraksi ke- Z ? Atraksi ke- Y akan terhitung pula. Tentunya, apabila $X = Y$, jawabannya ialah N .

Tugas

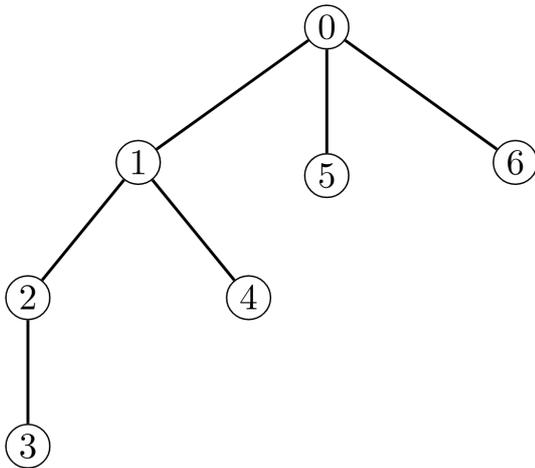
Anda harus mengimplementasi fungsi `createFunTour`:

- `createFunTour(N, Q)` - Fungsi ini akan dipanggil hanya sekali oleh *grader*.
 - N : Sebuah bilangan bulat yang menyatakan banyak atraksi.
 - Q : Sebuah bilangan bulat yang menyatakan jumlah pertanyaan maksimum.
 - Fungsi ini boleh memanggil dua fungsi juri:
 - `hoursRequired(X, Y)`
 - X : Sebuah bilangan bulat yang menyatakan atraksi pertama.
 - Y : Sebuah bilangan bulat yang menyatakan atraksi kedua.
 - Fungsi ini mengembalikan sebuah bilangan bulat yang menyatakan berapa jam yang diperlukan untuk berpindah dari atraksi ke- X menuju atraksi ke- Y .
 - Apabila X atau Y bukan sebuah bilangan bulat di antara 0 dan $N - 1$, Anda akan mendapatkan verdict WA.
 - `attractionsBehind(X, Y)`
 - X : Sebuah bilangan bulat yang menyatakan atraksi pertama.
 - Y : Sebuah bilangan bulat yang menyatakan atraksi kedua.

- Fungsi ini mengembalikan sebuah bilangan bulat yang menyatakan banyak atraksi Z sehingga Anda harus mengunjungi atraksi ke- Y saat Anda berpindah dari atraksi ke- X menuju atraksi ke- Z .
- Apabila X atau Y bukan sebuah bilangan bulat di antara 0 dan $N - 1$, Anda akan mendapatkan verdict WA.
- Fungsi ini harus mengembalikan sebuah *array* berisi N bilangan bulat yang menyatakan permutasi atraksi pada sebuah tur yang menyenangkan.

Contoh

Pada contoh berikut, $N = 7$, $Q = 400\,000$, $A = [0, 0, 0, 1, 1, 2]$, dan $B = [1, 5, 6, 2, 4, 3]$. Contoh tersebut diilustrasikan dengan gambar berikut:



Juri akan memanggil `createFunTour(7, 400000)`.

- Apabila kontestan memanggil `hoursRequired(3, 5)`, maka fungsi tersebut akan mengembalikan 4.
- Apabila kontestan memanggil `hoursRequired(5, 4)`, maka fungsi tersebut akan mengembalikan 3.
- Apabila kontestan memanggil `attractionsBehind(5, 1)`, maka fungsi tersebut akan mengembalikan 4. Untuk berpindah dari atraksi kelima ke atraksi pertama, kedua, ketiga, dan keempat, Anda harus mengunjungi atraksi pertama.
- Apabila kontestan memanggil `attractionsBehind(1, 5)`, maka fungsi tersebut akan mengembalikan 1.
- Kontestan dapat mengembalikan $[3, 6, 4, 5, 2, 0, 1]$ karena jam yang diperlukan untuk mengunjungi setiap atraksi selanjutnya berturut-turut ialah $[4, 3, 3, 3, 2, 1]$.

Batasan

- $2 \leq N \leq 100\,000$.
- $Q = 400\,000$.
- Dipastikan dapat berpindah antara setiap pasang atraksi melalui jalan-jalan tersebut.
- Setiap atraksi menjadi ujung dari paling banyak tiga jalan.

Subsoal 1 (10 poin)

- $N \leq 17$.

Subsoal 2 (16 poin)

- $N \leq 500$.

Subsoal 3 (21 poin)

- Ada sebuah jalan yang menghubungkan atraksi ke- i dan atraksi ke- $\lfloor \frac{i+1}{2} \rfloor$, untuk setiap $1 \leq i < N$.

Subsoal 4 (19 poin)

- Setidaknya ada sebuah atraksi T sehingga untuk setiap $0 \leq i < N$, berlaku `hoursRequired(T, i) < 30` dan terdapat sebuah selang $[L[i], R[i]]$ ($0 \leq L[i] \leq i \leq R[i] < N$) yang memenuhi kondisi-kondisi berikut:
 - Anda harus mengunjungi atraksi ke- i untuk berpindah dari atraksi ke- T menuju atraksi ke- j jika dan hanya jika $L[i] \leq j \leq R[i]$.
 - Jika $L[i] < i$, maka pasti terdapat tepat satu atraksi X sehingga:
 - $L[i] \leq X < i$.
 - Terdapat sebuah jalan yang menghubungkan atraksi ke- i dan atraksi ke- X .
 - Jika $i < R[i]$, maka pasti terdapat tepat satu atraksi Y sehingga:
 - $i < Y \leq R[i]$.
 - Terdapat sebuah jalan yang menghubungkan atraksi ke- i dan atraksi ke- Y .

Subsoal 5 (34 poin)

- Tidak ada batasan tambahan.

Contoh Grader

Contoh *grader* yang diberikan akan membaca masukan dengan format sebagai berikut:

```
N Q
A[0] B[0]
A[1] B[1]
.
.
.
A[N-2] B[N-2]
```

Contoh *grader* akan mencetak bilangan-bilangan bulat yang dikembalikan `createFunTour` apabila mengembalikan *array* berisi N bilangan bulat yang menyatakan permutasi dari atraksi-atraksi dalam tur yang menyenangkan serta memanggil `hoursRequired` dan `attractionsBehind` tidak lebih dari Q kali secara total. Selain itu, akan dicetak sebuah pesan *wrong answer*.