

# C. მხიარული ტური

Time limit	2 s
Memory limit	512 MB

## Description

ჯაკარტას ყველაზე დიდ თემატურ პარკში  $N$  ცალი ატრაქციონია, გადანომრილი 0-დან  $(N - 1)$ -მდე. ატრაქციონები დაკავშირებულია  $N - 1$  ორმხრივი გზით ისე, რომ ატრაქციონების ნებისმიერი წყვილისთვის უნიკალური დამაკავშირებელი მარშრუტი არსებობს. გზები დანომრილია 0-დან  $(N - 2)$ -მდე.  $i$ -ური გზა აკავშირებს  $A[i]$ -ურ და  $B[i]$ -ურ ატრაქციონებს და მის გავლას 1 საათი სჭირდება. საცობების თავიდან ასაცილებლად, თითოეული ატრაქციონი არის მაქსიმუმ 3 გზის ბოლო.

თქვენ გევალებათ შექმნათ ისეთი ტური, რომ თითოეულ ატრაქციონს ზუსტად ერთხელ ესტუმროთ. ატრაქციონიდან ატრაქციონამდე მისვლისას დიდი დროის დახარჯვა ძალიან მოსაწყენია. მხიარული ტურის შესაქმნელად თქვენ უნდა იპოვოთ ატრაქციონების ისეთი თანმიმდევრობა, რომ შემდეგ ატრაქციონამდე მისვლისთვის არ დაგჭირდეთ იმაზე მეტი დრო ვიდრე წინა ატრაქციონამდე მისვლისთვის. სხვა სიტყვებით რომ ვთქვათ, უნდა იპოვოთ მიმდევრობა  $P[0], P[1], \dots, P[N - 1]$ , რომელიც შეიცავს ყველა რიცხვს 0-დან  $(N - 1)$ -მდე ზუსტად ერთხელ, ისეთი, რომ დრო საჭირო  $P[i]$ -ური ატრაქციონიდან  $P[i + 1]$ -ურამდე მისვლისთვის არ აღემატება დროს, რომელიც საჭიროა  $P[i - 1]$ -ური ატრაქციონიდან  $P[i]$ -ურამდე მისვლისთვის, სადაც  $0 < i < N - 1$ .

თქვენ არ გაქვთ მოცემული ატრაქციონების სრული რუკა. მაშასადამე, საინფორმაციო ცენტრს უნდა დაუსვათ რამდენიმე შეკითხვა ტურის შესაქმნელად. თქვენ შეგიძლიათ მაქსიმუმ  $Q$  შეკითხვის დასმა, თითოეული  $X$  და  $Y$  პარამეტრებით, სადაც  $0 \leq X, Y < N$ . შეკითხვა შეიძლება იყოს 2 ტიპის:

- რამდენი საათია საჭირო  $X$ -ური ატრაქციონიდან  $Y$ -ურამდე მისაღწევად? თუ  $X = Y$ , პასუხი არის 0.
- რამდენი ისეთი  $Z$  ატრაქციონი არსებობს, რომ თქვენ მოგიწევთ  $Y$ -ური ატრაქციონის გავლა, თუ მივდივართ  $X$ -ურიდან  $Z$ -ურ ატრაქციონთან? პასუხში ჩაითვლება  $Y$ -ური ატრაქციონიც. თუ  $X = Y$ , პასუხი იქნება  $N$ .

## ამოცანა

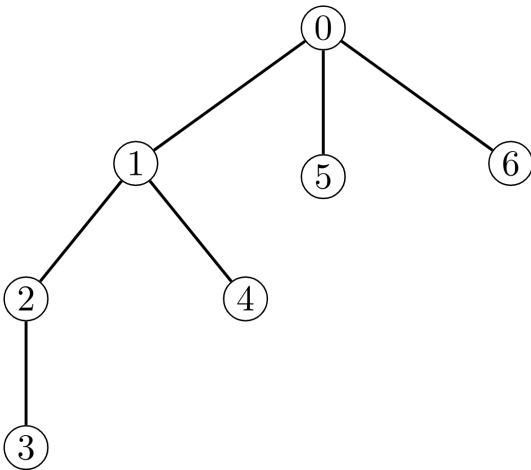
თქვენ უნდა მოახდინოთ ფუნქცია `createFunTour`-ის იმპლემენტაცია:

- `createFunTour(N, Q)` - ეს ფუნქცია გრადერის მიერ გამოცხადებული უნდა იყოს ზუსტად ერთხელ.
  - $N$ : მთელი რიცხვი, რომელიც წარმოადგენს ატრაქციონების რაოდენობას.
  - $Q$ : მთელი რიცხვი, რომელიც წარმოადგენს შეკითხვების მაქსიმალურ რაოდენობას.
  - ამ ფუნქციამ უნდა გამოიძახოს გრადერის ორი ფუნქცია:
    - `hoursRequired(X, Y)`
      - $X$ : მთელი რიცხვი, რომელიც წარმოადგენს პირველ ატრაქციონს.
      - $Y$ : მთელი რიცხვი, რომელიც წარმოადგენს მეორე ატრაქციონს.
      - ეს ფუნქცია აბრუნებს მთელ რიცხვს, რომელიც წარმოადგენს  $X$  ნომრის მქონე ატრაქციონიდან  $Y$  ნომრის მქონე ატრაქციონამდე მისასვლელი საათების საჭირო რაოდენობას.
      - თუ  $X$  ან  $Y$  არ არის მთელი რიცხვი 0-დან  $(N - 1)$ -მდე, თქვენ მიიღებთ ვერდიქტს WA.
    - `attractionsBehind(X, Y)`
      - $X$ : მთელი რიცხვი, რომელიც წარმოადგენს პირველ ატრაქციონს.
      - $Y$ : მთელი რიცხვი, რომელიც წარმოადგენს მეორე ატრაქციონს.

- ეს ფუნქცია აბრუნებს ინტეჯერს, რომელიც გვიჩვენებს ისეთი  $Z$  ატრაქციონების რაოდენობას, რომ  $Y$  ატრაქციონის გავლა მოგვინოს  $X$  ატრაქციონიდან  $Z$  ატრაქციონისკენ გზაზე.
- თუ  $X$  ან  $Y$  არ არის მთელი რიცხვი  $0$ -დან  $(N - 1)$ -მდე, თქვენ მიიღებთ ვერდიქტს WA.
- ამ ფუნქციამ უნდა დააბრუნოს  $N$  მთელი რიცხვისაგან შედგენილი მასივი, რომელიც წარმოადგენს ატრაქციონთა ნომრების პერმუტაციას მხიარული ტურში.

## მაგალითი

ქვემოთ მოყვანილ მაგალითში,  $N = 7$ ,  $Q = 400\,000$ ,  $A = [0, 0, 0, 1, 1, 2]$ , და  $B = [1, 5, 6, 2, 4, 3]$ . მაგალითი ილუსტრირებულია შემდეგ ნახაზზე:



გრადერი გამოიძახებს `createFunTour(7, 400000)`.

- თუ მონაწილე დასვამს შეკითხვას `hoursRequired(3, 5)`, ფუნქცია დააბრუნებს 4.
- თუ მონაწილე დასვამს შეკითხვას `hoursRequired(5, 4)`, ფუნქცია დააბრუნებს 3.
- თუ მონაწილე დასვამს შეკითხვას `attractionsBehind(5, 1)`, ფუნქცია დააბრუნებს 4. იმიათვის, რომ მეხუთე ატრაქციონიდან მიხვიდეთ პირველ, მეორე, მესამე და მეოთხე ატრაქციონებამდე, უნდა გაიაროთ პირველი ატრაქციონი.
- თუ მონაწილე დასვამს შეკითხვას `attractionsBehind(1, 5)`, ფუნქცია დააბრუნებს 1.
- მონაწილეს შეუძლია დაბრუნდეს `[3, 6, 4, 5, 2, 0, 1]`, რადგან საათები, რომელიც საჭიროა მომდევნო ატრაქციონების სანახავად თანმიმდევრულად ტოლია `[4, 3, 3, 3, 2, 1]`.

## შეზღუდვები

- $2 \leq N \leq 100\,000$ .
- $Q = 400\,000$ .
- ნებისმიერი ატრაქციონიდან არსებობს გზა ნებისმიერ სხვა ატრაქციონამდე.
- ყოველი ატრაქციონი წარმოადგენს საბოლოო წერტილს არაუმეტეს სამი გზისთვის.

## ქვეამოცანა 1 (10 ქულა)

- $N \leq 17$ .

## ქვეამოცანა 2 (16 ქულა)

- $N \leq 500$ .

## ქვეამოცანა 3 (21 ქულა)

- არსებობს გზა, რომელიც აერთებს  $i$ -ურ ატრაქციონს და  $\lfloor \frac{i-1}{2} \rfloor$ -ურ ატრაქციონს, ყველა  $1 \leq i < N$ -სათვის.

#### ქვეამოცანა 4 (19 ქულა)

- არსებობს მინიმუმ ერთი ატრაქციონი  $T$  ისეთი, რომ ყველა  $0 \leq i < N$ -სთვის, `hoursRequired(T, i)`  $< 30$  და არსებობს ინტერვალი  $[L[i], R[i]]$  ( $0 \leq L[i] \leq i \leq R[i] < N$ ) რომელიც შემდეგ პირობებს აკმაყოფილებს:
  - თქვენ უნდა ესტუმროთ  $i$ -ურ ატრაქციონს, რომ წახვიდეთ  $T$ -ური ატრაქციონიდან  $j$ -ურ ატრაქციონში მაშინ და მხოლოდ მაშინ როცა  $L[i] \leq j \leq R[i]$ .
  - თუ  $L[i] < i$ , მაშინ უნდა არსებობდეს ზუსტად ერთი ატრაქციონი  $X$  ისეთი, რომ:
    - $L[i] \leq X < i$ .
    - არსებობს გზა, რომელიც აერთებს  $i$  და  $X$  ატრაქციონებს.
  - თუ  $i < R[i]$ , მაშინ უნდა არსებობდეს ზუსტად ერთი ატრაქციონი  $Y$  ისეთი, რომ:
    - $i < Y \leq R[i]$ .
    - არსებობს გზა, რომელიც აერთებს  $i$  და  $Y$  ატრაქციონებს.

#### ქვეამოცანა 5 (34 ქულა)

- დამატებითი შეზღუდვების გარეშე.

#### სანიმუშო გრაფერი

სანიმუშო გრაფერი კითხულობს შესატან მონაცემებს შემდეგ ფორმატში:

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

სანიმუშო გრაფერი იწერს მთელ რიცხვებს, რომელსაც აბრუნებს `createFunTour` ფუნქცია, თუკი ის არის კორექტული  $N$ -ელემენტიანი მასივია და წარმოადგენს ატრაქციონთა პერმუტაციას მხიარული ტურის დროს და იძახებს `hoursRequired` და `attractionsBehind` ფუნქციებს არაუმეტეს  $Q$ -ჯერ ორივეს ერთად. წინააღმდეგ შემთხვევაში ის ბეჭდავს wrong answer შეტყობინებას.