



ყველაზე ხანგრძლივი მოგზაურობა

IOI 2023-ის ორგანიზატორები დიდი პრობლემის წინაშე აღმოჩნდნენ. მათ ხვალისათვის ოპუსტაბერში მოგზაურობის დაგეგმვა დაავიწყდათ. თუმცა, შეიძლება ჯერ კიდევ გვიანი არ იყოს.

ოპუსტაბერში N რაოდენობის ორიენტირია, რომლებიც გადანომრილია 0-დან $(N - 1)$ -მდე. ამ ორიენტირთა ზოგიერთი წყვილი *ორმხრივი გზებითაა* ერთმანეთთან დაკავშირებული, ხოლო თითოეული წყვილი ერთმანეთს არაუმეტეს ერთი გზით უკავშირდება. ორგანიზატორებმა არ იციან რომელი ორიენტირებია გზებით დაკავშირებული.

ვითყვი, რომ ოპუსტაბერში გზათა ქსელის *სიმკვრივე* არის *არანაკლებ δ* , თუ ყოველი 3 განსხვავებული ორიენტირისთვის არსებობს არანაკლებ δ რაოდენობის გზა მათ შორის. სხვა სიტყვებით, ორიენტირთა ყოველი ისეთი (u, v, w) სამეულისათვის, სადაც $0 \leq u < v < w < N$, ორიენტირთა (u, v) , (v, w) და (u, w) წყვილებიდან არანაკლებ δ რაოდენობის წყვილი გზებითაა ერთმანეთთან დაკავშირებული.

ორგანიზატორებმა იციან ისეთი მთელი დადებითი D რიცხვი, რომ გზათა სისტემის სიმკვრივე D -ზე არანაკლებია. ყურადღება მიაქციეთ იმას, რომ D -ს მნიშვნელობა არ შეიძლება 3-ზე მეტი იყოს.

ორგანიზატორებს შეუძლიათ ტელეფონით *დარეკონ* დისპეჩერს ოპუსტაბერში, რათა მიიღონ ინფორმაცია გარკვეულ ორიენტირებს შორის გზების არსებობის შესახებ. ყოველი დარეკვის დროს აუცილებელია მითითებული იქნას ორიენტირთა ორი არაყარეული $[A[0], \dots, A[P - 1]]$ და $[B[0], \dots, B[R - 1]]$ მასივი. ორიენტირები წყვილ-წყვილად განსხვავებული უნდა იყვნენ. ანუ:

- $A[i] \neq A[j]$ თითოეული ისეთი i -სათვის და j -სათვის, სადაც $0 \leq i < j < P$;
- $B[i] \neq B[j]$ თითოეული ისეთი i -სათვის და j -სათვის, სადაც $0 \leq i < j < R$;
- $A[i] \neq B[j]$ თითოეული ისეთი i -სათვის და j -სათვის, სადაც $0 \leq i < P$ და $0 \leq j < R$.

თითოეული დარეკვის დროს დისპეჩერი ამბობს, არსებობს თუ არა გზა, რომლითაც ერთმანეთთანაა დაკავშირებული ორიენტირები A -დან და B -დან. უფრო ზუსტად, დისპეჩერი განიხილავს i და j -ს ყველა ისეთ წყვილს, რომლისთვისაც $0 \leq i < P$ და $0 \leq j < R$. თუ, რომელიმე მათგანისათვის მაინც ორიენტირები $A[i]$ და $B[j]$ დაკავშირებული არიან გზით, მაშინ დისპეჩერი დააბრუნებს true-ს. წინააღმდეგ შემთხვევაში დისპეჩერი დააბრუნებს false-ს.

მოგზაურობა სიგრძით l არის ისეთი *განსხვავებული* $t[0], t[1], \dots, t[l - 1]$ ორიენტირთა მიმდევრობა, სადაც ყოველი i -სათვის, რომელიც მოთავსებულია 0-სა და $(l - 2)$ -ს შორის (ჩათვლით), ორიენტირები $t[i]$ და $t[i + 1]$ ერთმანეთთან გზით არიან დაკავშირებული.

მოგზაურობას სიგრძით l ეწოდება ყველაზე ხანგრძლივი მოგზაურობა, თუ არ არსებობს $l + 1$ ან მეტი სიგრძის არცერთი სხვა მოგზაურობა.

თქვენი ამოცანაა დაეხმაროთ ორგანიზატორებს იპოვონ ყველაზე ხანგრძლივი მოგზაურობა ოპუსტაბერში დისპენერთან დარეკვების საშუალებით.

იმპლემენტაციის დეტალები

თქვენ უნდა გაუკეთოთ იმპლემენტაცია შემდეგ პროცედურას:

```
int[] longest_trip(int N, int D)
```

- N : ორიენტირების რაოდენობა ოპუსტაბერში.
- D : გარანტირებული მინიმალური სიმკვრივე გზათა ქსელში.
- ამ პროცედურამ უნდა დააბრუნოს მასივი $t = [t[0], t[1], \dots, t[l - 1]]$, რომელიც წარმოადგენს ყველაზე ხანგრძლივ მოგზაურობას.
- ეს პროცედურა შეიძლება გამოძახებული იქნას რამდენჯერმე თითოეული ტესტისათვის.

გემოთ მოცემულ პროცედურას შეუძლია შემდეგი პროცედურების გამოძახება:

```
bool are_connected(int[] A, int[] B)
```

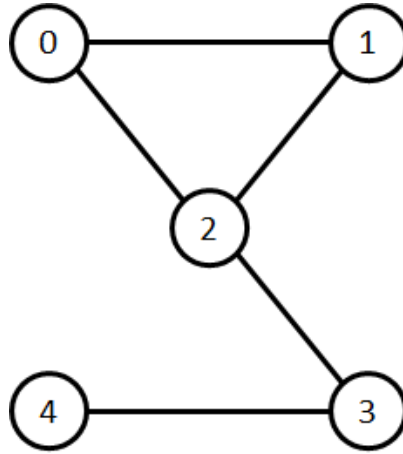
- A : განსხვავებული ორიენტირების არაცარიელი მასივი.
- B : განსხვავებული ორიენტირების არაცარიელი მასივი.
- A და B უნდა იყონ თანაკვეთი.
- ეს პროცედურა დააბრუნებს true-ს, თუ არსებობს ორიენტირი A -დან და ორიენტირი B -დან, რომლებიც დაკავშირებულია გზით. წინააღმდეგ შემთხვევაში დააბრუნებს false-ს.
- ეს პროცედურა შეიძლება იქნას გამოძახებული მაქსიმუმ 32 640-ჯერ longest_trip-ის თითოეული გამოძახებისთვის და მაქსიმუმ 150 000-ჯერ ჯამში.
- ჯამურმა სიგრძემ A და B მასივებისა, რომლებიც იქნება გადაცემული ამ პროცედურისთვის თითოეული გამოძახებისას, არ უნდა გადააჭარბოს 1 500 000-ს.

გრაფერი არ არის ადაპტიური. მნიშვნელობები N და D , ისევე როგორც წყვილები გზით დაკავშირებული ორიენტირებისა, იქნება დაფიქსირებული მანამ, სანამ longest_trip-ს გამოძახება მოხდება.

მაგალითები

მაგალითი 1

განვიხილოთ შემთხვევა როცა $N = 5$, $D = 1$ და გზები აკავშირებს ორიენტირებს ისე, როგორც სურათზეა მოცემული:



პროცედურა `longest_trip` არის გამოცხადებული შემდეგი ფორმით:

```
longest_trip(5, 1)
```

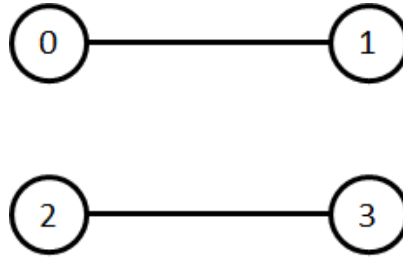
პროცედურას შეუძლია გამოიძახოს `are_connected` შემდეგნაირად.

გამოცხება	გზებით დაკავშირებული წყვილები	დაბრუნებული მნიშვნელობა
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) და (0,2)	true
<code>are_connected([2], [0])</code>	(2,0)	true
<code>are_connected([2], [3])</code>	(2,3)	true
<code>are_connected([1, 0], [4, 3])</code>	none	false

მეოთხე გამოცხების შემდეგ აღმოჩნდება, რომ (1,4), (0,4), (1,3) და (0,3) წყვილებიდან არცერთი არაა დაკავშირებული გზით. რადგან ქსელის სიმკვრივე არის მინიმუმ $D = 1$, ვხედავთ, რომ (0,3,4) სამეულიდან, წყვილი (3,4) უნდა იყოს დაკავშირებული გზით. ანალოგიურად, ორიენტირები 0 და 1 უნდა იყოს დაკავშირებული.

ამ მომენტში შეგვიძლია დავასკვნათ, რომ $t = [1, 0, 2, 3, 4]$ არის 5 სიგრძის მოგზაურობა და არსებობს 5-ზე მეტი სიგრძის მქონე მოგზაურობა. შესაბამისად, პროცედურა `longest_trip`-ს შეუძლია დააბრუნოს $[1, 0, 2, 3, 4]$.

განვიხილოთ სხვა შემთხვევა, როცა $N = 4$, $D = 1$ და გზები აკავშირებს ორიენტირებს ისე, როგორც სურათზეა მოცემული:



პროცედურა `longest_trip` გამოცხადებული იქნება შემდეგნაირად:

```
longest_trip(4, 1)
```

ასეთ შემთხვევაში ყველაზე ხანგრძლივი მოგზაურობა არის 2 სიგრძის. შესაბამისად, `are_connected` პროცედურის რამდენჯერმე გამოცხადების შემდეგ, პროცედურა `longest_trip`-ს შეუძლია დააბრუნოს ჩამოთვლილთაგან ერთ-ერთი $[0, 1]$, $[1, 0]$, $[2, 3]$ ან $[3, 2]$.

მაგალითი 2

ქვეამოცანა 0 შეიცავს დამატებით ტესტს $N = 256$ ორიენტირით. ტესტი შეგიძლიათ ნახოთ მიბმულ ფოლდერში, რომელიც შეიძლება გადმოწეროთ კონტენტის სისტემიდან.

შეზღუდვები

- $3 \leq N \leq 256$
- N -ების ჯამი `longest_trip`-ის ყველა გამოცხადებისთვის არ გადააჭარბებს 1 024-ს.
- $1 \leq D \leq 3$

ქვეამოცანები

1. (5 ქულა) $D = 3$
2. (10 ქულა) $D = 2$
3. (25 ქულა) $D = 1$. აღვნიშნოთ l^* -ით უგრძესი მოგზაურობის სიგრძე. პროცედურა `longest_trip`-ს არ მოეთხოვება დააბრუნოს მოგზაურობა სიგრძით l^* . ამის ნაცვლად, შეუძლია დააბრუნოს მოგზაურობა, რომლის სიგრძეც იქნება $\left\lceil \frac{l^*}{2} \right\rceil$ მაინც.
4. (60 ქულა) $D = 1$

თუ რომელიმე ტესტისათვის `are_connected` პროცედურის გამოცხადებები არ ეთანხმება იმპლემენტაციის დეტალებში მოცემულ შეზღუდვებს, ან `longest_trip`-ის მიერ დაბრუნებული მასივი არ არის სწორი, მაშინ თქვენი ქულა ამ ქვეამოცანაში იქნება 0.

მე-4 ქვეამოცანაში თქვენი ქულა დამოკიდებულია თქვენს მიერ `are_connected` პროცედურის გამოცხადების რაოდენობაზე `longest_trip`-ის ერთ შემთხვევაში. აღვნიშნოთ q -თი გამოყენებული გამოცხადებების მაქსიმალური რაოდენობა `longest_trip`-ის ყველა გამოცხადებას შორის ამ

ქვეამოცანის ყველა ტესტში. თქვენი ქულა ამ ქვეამოცანაში გამოითვლება შემდეგი ცხრილის მიხედვით:

შეზღუდვა	ქულები
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

სანიმუშო გრაფერი

C -თი აღვნიშნოთ შემთხვევების რაოდენობა, ანუ ტესტში `longest_trip`-ის გამოძახებების რაოდენობა. სანიმუშო გრაფერი კითხულობს შესატან მონაცემებს შემდეგნაირად:

- ხაზი 1: C

ამას მოყვება C ცალი შემთხვევის აღწერა.

სანიმუშო გრაფერი კითხულობს თითოეულ შემთხვევას შემდეგნაირად:

- ხაზი 1: $N\ D$
- ხაზი $1 + i$ ($1 \leq i < N$): $U_i[0]\ U_i[1]\ \dots\ U_i[i - 1]$

აქ თითოეული U_i ($1 \leq i < N$) i ზომის მასივია, რომელიც აღწერს ორიენტირების წყვილებს, რომლებიც დაკავშირებულია გზებით. თითოეული i და j -სთვის, რომელთათვისაც $1 \leq i < N$ და $0 \leq j < i$:

- თუ ორიენტირები j და i არიან დაკავშირებული გზით, $U_i[j]$ -ს მნიშვნელობა უნდა იყოს 1;
- თუ ორიენტირები j და i არიან დაკავშირებული გზით, $U_i[j]$ -ს მნიშვნელობა უნდა იყოს 0.

თითოეულ შემთხვევაში, `longest_trip`-ის გამოძახებამდე, სანიმუშო გრაფერი ამოწმებს არის თუ არა ქსელის სიმკვრივე მინიმუმ D . თუ ეს პირობა არ იქნება დაკმაყოფილებული, მაშინ გრაფერი გამოიტანს შეტყობინებას `Insufficient Density` და შეწყვეტს მუშაობას.

თუ სანიმუშო გრაფერი დააფიქსირებს პროტოკოლის დარღვევას, ის გამოიტანს შეტყობინებას `Protocol Violation: <MSG>`, სადაც `<MSG>` იქნება ერთ-ერთი შემდგომ შეცდომებს შორის:

- `invalid array: are_connected` გამოძახებისას, A და B მასივებიდან ერთ-ერთი მაინც არის
 - ცარიელი, ან
 - შეიცავს ელემენტს, რომელიც არ არის მთელი რიცხვი 0-დან $(N - 1)$ -ის ჩათვლით ინტერვალში, ან
 - შეიცავს რომელიმე ელემენტს მინიმუმ 2-ჯერ.

- non-disjoint arrays: `are_connected`, A და B მასივები არ არიან თანაუკვეთი.
- too many calls: `are_connected` გამოძახებების რაოდენობა აჭარბებს 32 640-ს `longest_trip`-ის მიმდინარე გამოძახებაში, ან აჭარბებს 150 000-ს ჯამში.
- too many elements: `are_connected`-სთვის გადაცემული ორიენტირების ჯამური რაოდენობა აჭარბებს 1 500 000-ს.

წინააღმდეგ შემთხვევაში, რაღაც არაუარყოფითი l -სთვის `longest_trip`-ის მიმდინარე შემთხვევის მიერ დაბრუნებული მასივის ელემენტები აღვნიშნოთ $t[0], t[1], \dots, t[l-1]$ -ით. სანიმუშო გრაფერი დაბეჭდავს 3 სტრიქონს შემდეგი ფორმატით:

- ხაზი 1: l
- ხაზი 2: $t[0] \ t[1] \ \dots \ t[l-1]$
- ხაზი 3: `are_connected` გამოძახებების რაოდენობა ამ შემთხვევაში.

საბოლოოდ სანიმუშო გრაფერი ბეჭდავს:

- ხაზი $1 + 3 \cdot C$: `are_connected` გამოძახებების მაქსიმალური რაოდენობა `longest_trip`-ის ყველა გამოძახებას შორის.