



Ən uzun səyahət

IOI 2023'ün organizatorları çətin vəziyyətə düşüblər. Onlar növbəti günlər üçün Ópusztaszer kəndinə səyahət planlamağı unudublar. Amma bəlkə hələ də çox gec deyil...

Ópusztaszer'da 0'dan $N - 1$ -ə nömrələnmiş N sayda görülməli məkan var. Bu yerlərin bəziləri *iki istiqamətli yollarla* əlaqələndirilib. Hər bir məkan cütü ən çox bir yol ilə əlaqələndirilib. Organizatorlar hansı yerlər bir-biri ilə əlaqələndirilib *bilmirlər*.

Ópusztaszer'da olan yol şəbəkəsində istənilən 3 müxtəlif məkan arasında ən az δ sayda yol varsa, bu şəbəkənin **sıxlığına** δ deyək. Başqa sözlə, $0 \leq u < v < w < N$ şərtini ödəyən istənilən (u, v, w) üçlüyü üçün, (u, v) , (v, w) və (u, w) cütləri arasında ən az δ sayda cüt yol ilə əlaqələndirilib.

Organizatorlar elə bir D ədədi *bilir* ki, bu şəbəkənin sıxlığı ən az D 'dir. Nəzər yetirin ki, D 3'dən böyük ola bilməz.

Organizatorlar Ópusztaszer'a **zənglər** edərək bəzi məkanlar arasındakı yollar barədə məlumatlar ala bilirlər. Hər bir zəngdə iki boş olmayan $[A[0], \dots, A[P - 1]]$ və $[B[0], \dots, B[R - 1]]$ məkan massivləri verilməlidir. Bütün məkanlar müxtəlif olmalıdır, yəni

- $A[i] \neq A[j]$, $0 \leq i < j < P$ şərtini ödəyən hər i və j üçün;
- $B[i] \neq B[j]$, $0 \leq i < j < R$ şərtini ödəyən hər i və j üçün;
- $A[i] \neq B[j]$, $0 \leq i < P$ və $0 \leq j < R$ şərtini ödəyən hər i və j üçün.

Hər zəngə cavab olaraq A və B arasında əlaqə olub olmadığı deyilir. Daha dəqiq, bütün i, j ($0 \leq i < P$, $0 \leq j < R$) cütləri üçün $A[i]$ və $B[j]$ arasında yol olub-olmadığına baxılır və əgər hər hansı $A[i]$ və $B[j]$ arasında yol olarsa, o zaman `true`, əks halda `false` cavabı qaydır.

l uzunluqlu $t[0], t[1], \dots, t[l - 1]$ məkan ardıcılığına o vaxt **səyahət** deyək ki, bütün məkanlar müxtəlif olsun, 0 və $l - 2$ arasında olan hər i üçün $t[i]$ və $t[i + 1]$ məkanları arasında yol olsun. l uzunluqlu səyahət o zaman **ən uzun səyahət** adlansın ki, $l + 1$ uzunluqlu səyahət olmasın.

Sizin vəzifəniz Ópusztaszer'da olan ən uzun səyahəti tapmaqda organizatorlara kömək etməkdir.

İmplementasiya detalları

Aşağıdakı proseduru implement etməlisiniz:

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

- N : Ópusztaszer'da olan məkanların sayı.
- D : yol şəbəkəsinin zəmanət verilən ən az sıxlığı
- Bu prosedur ən uzun səyahəti göstərən $t = [t[0], t[1], \dots, t[l - 1]]$ massivini qaytarmalıdır.
- Bu prosedur hər test zamanı **bir neçə dəfə çağırılmalıdır**.

Yuxarıda verilən prosedur aşağıdakı proseduru çağırır:

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

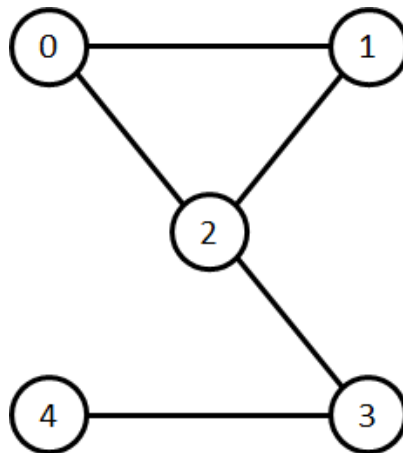
- A : boş olmayan müxtəlif məkan massivi.
- B : boş olmayan müxtəlif məkan massivi.
- A və B massivinın ortaq elementi yoxdur.
- Əgər A 'dakı hər hansı məkan ilə B 'dəki hər hansı məkan arasında yol varsa bu zaman prosedur true, əks halda false qaytarır.
- Bu prosedur hər longest_trip proseduru zamanı ən çox 32 640 dəfə çağırılmalıdır, ümumilikdə isə ən çox 150 000 dəfə çağırılmalıdır.
- A və B massivinın ümumi uzunluqlarının cəmi 1 500 000'i keçməməlidir.

Qreyder adaptiv **deyil**. Hər bir həll eyni testlər üzərində yoxlanılır. Yəni N və D dəyərləri, həmçinin hansı məkanlar arasında yollar olduğu hər bir test daxilində longest_trip prosedurunun hər bir çağırışı üçün dəyişməzdir.

Nümunələr

Nümunə 1

$N = 5$, $D = 1$, və yolların aşağıdakı kimi qoşulduğu bir ssenariyə nəzər yetirək:



longest_trip proseduru aşağıdakı kimi çağırılır:

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

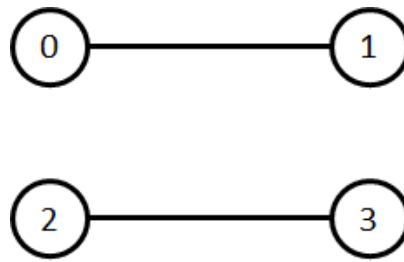
Bu prosedur are_connected proseduruna aşağıdakı kimi çağırışlar edə bilər.

Çağırış	Qoşulu olan yollar	Qaytarılan dəyər
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) və (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>	heç biri	false

Dördüncü çağırışda məlum olur ki (1,4), (0,4), (1,3) və (0,3) cütlərinin *heç biri* arasında yol yoxdur. Şəbəkənin sıxlığı ən az $D = 1$ olduğuna görə, (0,3,4) üçlüyündən (3,4) cütü arasında yol olmalıdır. Oxşar şəkildə, 0 və 1 məkanları arasında da yol olmalıdır.

Burada artıq 5 uzunluqlu $t = [1, 0, 2, 3, 4]$ ardıcılığının ən uzun səyahət olduğunu deyə bilirik, və daha uzun səyahət tapmaq mümkün deyil. Buna görə də `longest_trip` proseduru $[1, 0, 2, 3, 4]$ qaytara bilər.

$N = 4$, $D = 1$, və yolların aşağıdakı kimi qoşulduğu başqa bir ssenariyə baxaq:



`longest_trip` proseduru aşağıdakı kimi çağırılır:

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

Burada ən uzun səyahətin uzunluğu 2'dir. Buna görə də `are_connected`-ə olan bir neçə çağırışdan sonra `longest_trip` proseduru $[0, 1]$, $[1, 0]$, $[2, 3]$ or $[3, 2]$ cütlərindən hər hansı birini qaytara bilər.

Nümunə 2

0'cı alt tapşırıqında əlavə olaraq $N = 256$ olan test var. Bu testi yarış sistemində olan yükləyə bildiyiniz əlavənin içində görə bilərsiniz.

Məhdudiyyətlər

- $3 \leq N \leq 256$
- `longest_trip`-ə olan çağırışlarda N -lərin cəmi 1 024-ü keçmir.
- $1 \leq D \leq 3$

Alt tapşırıqlar

1. (5 bal) $D = 3$
2. (10 bal) $D = 2$
3. (25 bal) $D = 1$. l^* ən uzun səyahətin uzunluğunu göstərsin. `longest_trip` proseduru l^* uzunluqlu səyahət qaytarmaq məcburiyyətində deyil. Əvəzində ən az $\left\lceil \frac{l^*}{2} \right\rceil$ uzunluqlu səyahət qaytarmağı kifayətdir.
4. (60 bal) $D = 1$

4'cü alt tapşırıqda sizin balınız bir `longest_trip` proseduru zamanı `are_connected` proseduruna olan çağırışların sayından asılı olacaq. Alt tapşırığın hər testindəki bütün `longest_trip` çağırışları zamanı `are_connected` maksimum q dəfə çağırılmış olsun. Sizin balınız aşağıdakı cədvələ uyğun hesablanacaq.

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

Əgər hər hansı test'də `are_connected` proseduruna olan çağırışlar `İmplementasiya Detallarında` verilən şərtlərə uyğun gəlməsə və ya `longest_trip`'in qaytardığı cavab səhv olarsa, uyğun alt tapşırıq 0 bal verəcək.

Nümunə qreyder

C ssenarilərin sayını, yəni `longest_trip` proseduruna olan çağırışların sayını göstərsin. Nümunə qreyder girişi aşağıdakı formada oxuyur:

- sətir 1: C

C ssenarinin izahı ilə giriş davam edir.

Nümunə qreyder hər ssenarinin izahını aşağıdakı formatda oxuyur:

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

Burada hər bir U_i ($1 \leq i < N$) i ölçülü bir massivdir və məkanlar arasındakı yolları göstərir. $1 \leq i < N$ və $0 \leq j < i$ şərtini ödəyən hər bir i və j üçün:

- əgər j və i məkanı arasında yol varsa o zaman $U_i[j]$ dəyəri 1 olmalıdır;
- əgər j və i məkanı arasında yol yoxdursa o zaman $U_i[j]$ dəyəri 0 olmalıdır.

Hər bir ssenaridə `longest_trip` çağırılmazdan əvvəl nümunə qreyder yol şəbəkəsinin sıxlığının ən az D olub olmadığını yoxlayır. Əgər şərt ödənməsə `Insufficient Density` mesajı göstərir və dayanır.

Əgər nümunə qreyder protokoldan kənara çıxma təyin edərsə o zaman çıxışa `Protocol Violation: <MSG>` verir. Burada `<MSG>` aşağıdakı mesajlardan biridir:

- `invalid array: are_connected` proseduruna olan çağırışların birində, ya A ya da B
 - boşdur, və ya
 - 0 və $N - 1$ arasında olmayan bir ədəd var, və ya
 - hansısa elementdən orda ən az iki ədəd var.
- `non-disjoint arrays: are_connected` çağırılması üçün istifadə edilən A və B massivlərinin ortaq elementi var.
- `too many calls`: bir `longest_trip` çağırışı zaman `are_connected`-a olan çağırışların sayı $32\,640$ -ı və ya ümumi limit olan $150\,000$ -i aşır.
- `too many elements`: `are_connected`-a olan bütün çağırışlarda istifadə olunan massivlərdəki elementlərin sayları cəmi $1\,500\,000$ -i keçir.

Əks halda, `longest_trip` prosedurunun qaytardığı massiv hansısa mənfi olmayan l ədədi üçün $t[0], t[1], \dots, t[l - 1]$ olsun. Nümunə qreyder bu ssenari üçün aşağıdakı formatda üç sətir çıxışa verir:

- sətir 1: l
- sətir 2: $t[0] \ t[1] \ \dots \ t[l - 1]$
- sətir 3: bu ssenaridə `are_connected` proseduruna olan çağırışların sayı

Ən sonda nümunə qreyder aşağıdakı məlumatı çıxışa verir:

- sətir $1 + 3 \cdot C$: bütün `longest_trip` prosedurları arasında `are_connected` proseduruna olan maksimum çağırışların sayı