



# Најдужи пут

Организатори ИОИ 2023 су у велико проблему! План екскурзије у Горњи Брестовац је украден од стране мрачног Милисава! Али можда још није касно...

Постоји  $N$  атракција у Горњем Брестовцу, и оне су индексирани од 0 до  $N - 1$ . Неки парови атракција су повезани *двосмерним путевима*. Сваки пар атракција је повезан највише једним путем. Организатори *не знају* које атракције су повезане путевима.

Кажемо да је **густина** мреже путева у Горњем Брестовцу **најмање**  $\delta$  ако сваке 3 различите атракције имају барем  $\delta$  путева између њих. Другим речима, за сваку тројку атракција  $(u, v, w)$  тако да је  $0 \leq u < v < w < N$ , међу паровима атракција  $(u, v)$ ,  $(v, w)$  и  $(u, w)$  барем  $\delta$  парова су повезани путем.

Организатори *знају* природан број  $D$  такав да је густина мреже путева барем  $D$ . Приметимо да вредност  $D$  не може бити већа од 3.

Организатори могу да **зову** селоначелника Горњег Брестовца како би сазнали информације о повезаности између неких атракција. У сваком позиву, два непразна низа атракција  $[A[0], \dots, A[P - 1]]$  и  $[B[0], \dots, B[R - 1]]$  морају да буду достављена. Атракције морају да буду различите, то јест:

- $A[i] \neq A[j]$  за свако  $i$  и  $j$  тако да је  $0 \leq i < j < P$ ;
- $B[i] \neq B[j]$  за свако  $i$  и  $j$  тако да је  $0 \leq i < j < R$ ;
- $A[i] \neq B[j]$  за свако  $i$  и  $j$  тако да је  $0 \leq i < P$  и  $0 \leq j < R$ .

За сваки позив, селоначелник одговара да ли постоји пут који повезује атракцију из  $A$  и атракцију из  $B$ . То јесте, селоначелник враћа `true` уколико постоје  $i$  и  $j$  такви да  $0 \leq i < P$  и  $0 \leq j < R$ , и  $A[i]$  и  $B[j]$  су повезани путем. Ако не постоје, селоначелник враћа `false`.

**Путовање** дужине  $l$  је низ *различитих* атракција  $t[0], t[1], \dots, t[l - 1]$ , где за свако  $i$  од 0 до  $l - 2$ , атракција  $t[i]$  и атракција  $t[i + 1]$  су повезане путем. Путовање дужине  $l$  зовемо **најдужим путовањем** уколико не постоји ниједно путовање дужине барем  $l + 1$ .

Ваш задатак је да помогнете организаторима да нађу најдуже путовање кроз Горњи Брестовац, користећи позиве ка селоначелнику.

## Детаљи имплементације

Треба да имплементирате следећу процедуру:

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

- $N$ : број атракција у Горњем Брестовцу.
- $D$ : гарантовања минимална густина мреже путева.
- Ова процедура треба да врати низ  $t = [t[0], t[1], \dots, t[l-1]]$ , који представља најдуже путовање.
- Ова процедура може бити позвана **више пута** у сваком тест примеру.

Ова процедура може да позива следећу процедуру:

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

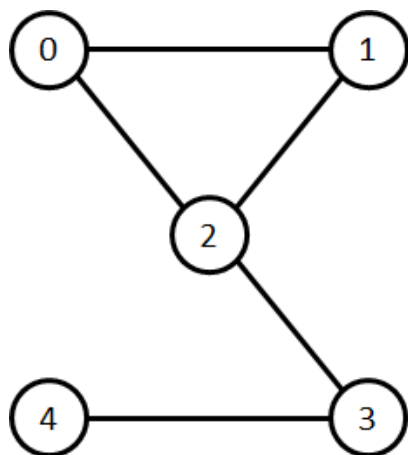
- $A$ : непразан низ различитих атракција.
- $B$ : непразан низ различитих атракција.
- $A$  и  $B$  морају да буду дисјунктни.
- Ова процедура враћа `true` уколико постоји атракција из  $A$  и атракција из  $B$  које су повезане путем. У супротном, враћа `false`.
- Ова процедура може да буде позвана највише 32 640 пута при сваком позиву функције `longest_trip`, и највише 150 000 пута укупно.
- Укупна дужина низова  $A$  и  $B$  прослеђених овој функцији, по свим позивима, не може да премаши 1 500 000.

Оцењивач (grader) **није адаптиван**. Вредности  $N$  и  $D$ , као и парови атракција који су повезани путем, су фиксирани пре него што се функција `longest_trip` позове.

## Примери

### Пример 1

Посматрајмо сценарио у ком је  $N = 5$ ,  $D = 1$ , и путеви су као на следећој слици:



Процедура `longest_trip` је позвана на следећи начин:

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

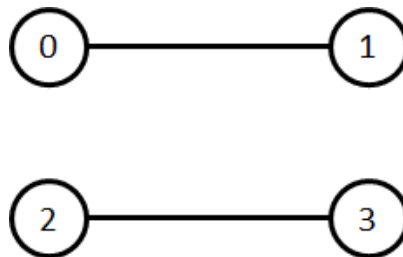
Ова процедура може да позива процедуру `are_connected` на следећи начин.

Позив	Парови повезани путем	Повратна вредност
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) и (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>	ништа	false

После четвртог позива, испоставља се да *ниједан* од парова (1,4), (0,4), (1,3) и (0,3) није повезан путем. Како је густина мреже најмање  $D = 1$ , можемо да приметимо да из тројке (0,3,4) пар (3,4) мора бити повезан путем. Слично, атракције 0 и 1 морају да буду повезане.

У том тренутку, можемо да закључимо да  $t = [1, 0, 2, 3, 4]$  је путовање дужине 5, и да не постоји путовање дужине преко 5. Дакле, процедура `longest_trip` може да врати `[1, 0, 2, 3, 4]`.

Посматрајмо други сценарио у коме је  $N = 4$ ,  $D = 1$ , и путеви између атракција су као на следећој слици:



Процедура `longest_trip` је позвана на следећи начин:

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

У овом сценарију, дужина најдужег путовања је 2. Дакле, након пар позива процедуре `are_connected`, процедура `longest_trip` може да врати нешто од `[0, 1]`, `[1, 0]`, `[2, 3]` или `[3, 2]`.

## Пример 2

Подзадатак 0 садржи додатни пример "са папира" са  $N = 256$  атракција. Овај тест пример је укључен у закачку која се може скинути са система за такмичење.

## Ограничења

- $3 \leq N \leq 256$
- Збир  $N$ -ова по свим позивима функције `longest_trip` не премашује 1 024.
- $1 \leq D \leq 3$

## Подзадаци

1. (5 поена)  $D = 3$
2. (10 поена)  $D = 2$
3. (25 поена)  $D = 1$ . Нека је  $l^*$  дужина најдужег путовања. Процедуре `longest_trip` не мора да врати путовање дужине  $l^*$ . Уместо тога, треба да врати путовање дужине барем  $\left\lceil \frac{l^*}{2} \right\rceil$ .
4. (60 поена)  $D = 1$

Ако, у било ком тест примеру, позиви ка процедури `are_connected` не одговарају условима описаним у детаљима имплементације, или ако је низ враћен од функције `longest_trip` нетачан, резултат вашег решења за тај подзадатак ће бити 0.

У подзадатку 4 ваш скор је одређен бројем позива процедуре `are_connected` по једном позиву процедуре `longest_trip`. Нека је  $q$  максималан број позива по свим инвокацијама процедуре `longest_trip` по сваком тест примеру у подзадатку. Ваш скор за тај подзадатак је израчунат по следећој табели:

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

## Пример оцењивача (sample grader)

Нека је  $C$  број сценарија, то јест, број позива `longest_trip`. Оцењивач чита улаз у следећем формату:

- линија 1:  $C$

Описи  $C$  сценарија следе.

Оцењивач чита опис сваког сценарија у следећем формату:

- линија 1:  $N\ D$

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

Овде, свако  $U_i$  ( $1 \leq i < N$ ) је низ дужине  $i$ , који описује који парови атракција су повезани путем. За свако  $i$  и  $j$  тако да важи  $1 \leq i < N$  и  $0 \leq j < i$ :

- ако су атракције  $j$  и  $i$  повезане путем, онда вредност  $U_i[j]$  треба да буде 1;
- ако не постоји пут који повезује атракције  $j$  и  $i$ , онда вредност  $U_i[j]$  треба да буде 0.

У сваком сценарију, пре позива `longest_trip`, оцењивач проверава да ли је густина мреже путева најмање  $D$ . Ако овај услов није испуњен, оцењивач исписује `Insufficient Density` и завршава.

Ако оцењивач примети кршење протокола, испис оцењивача је: `Protocol Violation: <MSG>`, где је `<MSG>` једна од следећих ствари:

- `invalid array`: у позиву `are_connected`, барем један од низова  $A$  и  $B$ 
  - је празан, или
  - садржи елемент који није број од 0 до  $N - 1$ , или
  - садржи исти елемент најмање двапут.
- `non-disjoint arrays`: у позиву `are_connected`, низови  $A$  и  $B$  нису дисјунктни.
- `too many calls`: број позива процедуре `are_connected` премашује 32 640 у тренутном позиву `longest_trip`, или премашује 150 000 укупно.
- `too many elements`: укупан број атракција прослеђен процедури `are_connected` по свим позивима премашује 1 500 000.

У супротном, нека су елементи низа враћени од стране `longest_trip` у сценарију  $t[0], t[1], \dots, t[l - 1]$  за неко ненегативно  $l$ . Оцењивач исписује три линије за сценарио, у следећем формату:

- линија 1:  $l$
- линија 2:  $t[0] \ t[1] \ \dots \ t[l - 1]$
- линија 3: број позива процедуре `are_connected` у овом сценарију

На крају, оцењивач исписује:

- линија  $1 + 3 \cdot C$ : максималан број позива процедуре `are_connected` по свим позивима процедуре `longest_trip`