



Longest Trip

IOI 2023 ұйымдастырушылары үлкен қиындыққа тап болды! Олар алдағы Ópusztaszer-ге сапарды жоспарлауды ұмытып кетті. Бірақ әлі де кеш емес шығар...

Ópusztaszer-де 0-дан $N - 1$ -ге дейін нөмірленген N көрнекі жерлері бар. Бұл көрнекі жерлердің кейбіреуі *қос бағытты жолдар* арқылы қосылған. Көрнекі жерлердің әрбір жұбы ең көбі бір жол арқылы қосылады. Ұйымдастырушылар қандай көрнекі жерлер жолдармен байланысты екенің білмейді.

Біз Ópusztaszer жол желісінің **тығыздығы кем дегенде δ** деп айтамыз, егер әрбір 3 әр түрлі көрнекі жерлердің арасында кемінде δ жолдар болса. Басқаша айтқанда, $0 \leq u < v < w < N$ болатындай кез келген (u, v, w) үштігі үшін (u, v) , (v, w) және (u, w) жұптарының арасында кем дегенде δ жол болу керек.

Ұйымдастырушылар жол желісінің тығыздығы кемінде D болатындай, D оң бүтін санын біледі. D мәні 3-тен көп болмайтының ескеріңіз.

Ұйымдастырушылар белгілі бір көрнекті жерлер арасындағы жол байланыстары туралы ақпарат жинау үшін Ópusztaszer диспетчеріне **қоңырау** жасай алады. Әрбір қоңырауда $[A[0], \dots, A[P - 1]]$ және $[B[0], \dots, B[R - 1]]$ көрнекі орындардың бос емес екі массиві көрсетілуі керек

Қоңыраудағы көрнекті жерлер әр түрлі болуы қажет, яғни

- $0 \leq i < j < P$ болатын әрбір i және j үшін $A[i] \neq A[j]$;
- $0 \leq i < j < R$ болатын әрбір i және j үшін $B[i] \neq B[j]$;
- $0 \leq i < P$ және $0 \leq j < R$ болатын әрбір i және j үшін $A[i] \neq B[j]$.

Әрбір қоңырау үшін диспетчер A көрнекті орындарын B көрнекті орындарымен байланыстыратын жолдың бар-жоғын хабарлайды. Яғни, $0 \leq i < P$, $0 \leq j < R$ және $A[i]$ мен $B[j]$ арасында жол болатын i және j табылса, диспетчер "true" қайтарады. Әйтпесе, диспетчер «false» мәнін қайтарады.

l ұзындықты **саяхат** деп $t[0], t[1], \dots, t[l - 1]$ әр түрлі көрнекі жерлер тізбегін айтамыз, мұндағы 0 және $l - 2$ арасындағы әрбір i үшін, $t[i]$ көрнекі жері және $t[i + 1]$ көрнекі жері жол арқылы қосылған. Ұзындығы l болатын саяхат **ең ұзақ саяхат** деп аталады, егер ұзындығы кемінде $l + 1$ болатын саяхат табылмаса.

Сіздің міндетіңіз - диспетчерге қоңырау шалу арқылы ұйымдастырушыларға Ópusztaszer-те ең ұзақ саяхатты табуға көмектесу.

Implementation Details

Сізге келесі функцияны асыру қажет:

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

- N : Ópusztaszerдегі көрнекі жерлер саны.
- D : жол желісінің кепілдік берілген ең төменгі тығыздығы.
- Бұл функция ең ұзақ саяхатты сипаттайтын $t = [t[0], t[1], \dots, t[l - 1]]$ массивын қайтару қажет.
- Бұл функция әр тестте **бірнеше рет** шақыртылады.

Үстіндегі функция келесі функцияны шақыра алады:

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

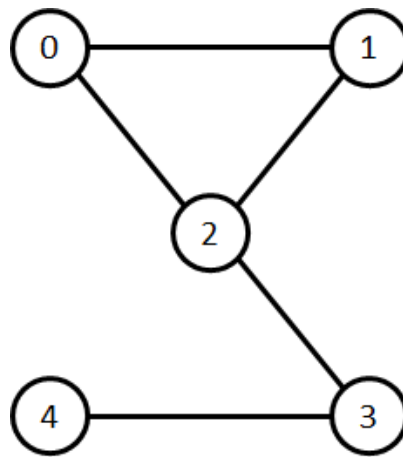
- A : әр түрлі көрнекі жерлердің массиві
- B : әр түрлі көрнекі жерлердің массиві.
- A және B қиысыспайтын жиындар болуы қажет.
- Бұл функция true қайтарады егер A дағы көрнекі жер мен B дағы көрнекі жер арасында жол болса. Болмаса, ол false қайтарады.
- Әрбір longest_trip ішінде бұл функция 32 640 көп шақырмалуы қажет, және барлығы 150 000-нен көп емес.
- Осы функцияға оның барлық шақыруларында берілген A және B массивтерінің жалпы ұзындығы 1 500 000-дан аспауы керек.

Грейдер **бейімделген**. N және D мәндері, сондай-ақ жолдармен байланыстырылған көрнекі жерлер жұптары «longest_trip» функциясын шақыру алдында бекітіледі.

Examples

Example 1

$N = 5$, $D = 1$ және жол байланыстары келесі суретте көрсетілгендей болатын сценарийді қарастырыңыз:



Функция `longest_trip` келесі жолмен шақырылады:

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

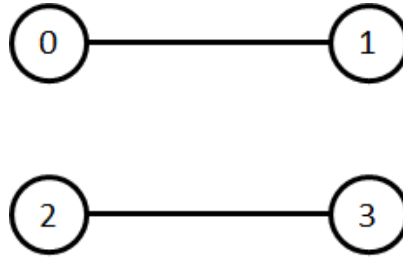
Функция `are_connected` келесі жолдармен шақырылуы мүмкін.

| Қоңырау | Жолмен қосылған жұптар | Қайтарылатын мән |
|---|------------------------|------------------|
| <code>are_connected([0], [1, 2, 4, 3])</code> | (0,1) және (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> | ешқайсысы | false |

Төртінші қоңыраудан кейін (1,4), (0,4), (1,3) және (0,3) жұптарының *ешқайсысы* жолмен қосылмағаны белгілі болды. Желінің тығыздығы кем дегенде $D = 1$ болғандықтан, (0,3,4) үштігінде (3,4) жұбы жол арқылы қосылуы керек екенін көреміз. Осыған ұқсас 0 және 1 көрнекі жерлері қосылуы керек.

Осы кезде $t = [1, 0, 2, 3, 4]$ ұзындығы 5 болатын саяхат және 5-тен асатын саяхат жоқ деген қорытынды жасауға болады. Сондықтан, «longest_trip» процедурасы $[1, 0, 2, 3, 4]$ қайтаруы мүмкін.

$N = 4$, $D = 1$ және бағдарлар арасындағы жолдар келесі суретте көрсетілгендей болатын басқа сценарийді қарастырыңыз:



Функция `longest_trip` келесі жолмен шақырылады:

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

Бұл сценарийде ең ұзақ сапардың ұзақтығы 2 құрайды. Сондықтан, 'are_connected' процедурасына бірнеше шақырудан кейін `longest_trip` процедурасы `[0, 1]`, `[1, 0]`, `[2, 3]` немесе `[3, 2]` мәндерінің бірін қайтаруы мүмкін.

Example 2

0ші бөлімшеде $N = 256$ бар қосымша тест мысалы бар. Бұл тест контекст жүйесінен жүктеп алуға болатын қосымша архивте бар.

Constraints

- $3 \leq N \leq 256$
- Барлық `longest_trip` шақырылуында N қосындысы 1 024-тен аспайды.
- $1 \leq D \leq 3$

Subtasks

1. (5 points) $D = 3$
2. (10 points) $D = 2$
3. (25 points) $D = 1$. l^* деп ең ұзақ саяхаттың ұзындығын айтайық. Функция `longest_trip` l^* қайтаруы кажет емес. Оның орнына ол ұзындығы $\left\lceil \frac{l^*}{2} \right\rceil$ кем емес болатын саяхатты қайтаруға болады.
4. (60 points) $D = 1$

Егер тесттердің кез келгенінде «are_connected» функциясын шақыру кезінде сипатталған шектеулерге сәйкес келмесе немесе «longest_trip» арқылы қайтарылған массив дұрыс емес болса, сол бөлімше үшін шешіміңіздің ұпайы 0 болады.

4-ші бөлімшеде сіздің ұпайыңыз «longest_trip» функциясы «are_connected» функциясын шақырулар саны негізінде анықталады. q - бөлімше тесттері арасында "longest_trip" шақыруларының ішіндегі ең көп қоңыраулар саны болсын. Осы бөлімше бойынша ұпайыңыз келесі кестеге сәйкес есептеледі:

| Condition | Points |
|---------------------------|--------|
| $2\,750 < q \leq 32\,640$ | 20 |
| $550 < q \leq 2\,750$ | 30 |
| $400 < q \leq 550$ | 45 |
| $q \leq 400$ | 60 |

Sample Grader

C деп тесттер саның айтайық, яғни, `longest_trip` функциясының шақыртулар саны. Үлгі бағалаушы кірісті келесі форматта оқиды:

- жол 1: C

C тесттер саны.

Үлгі бағалаушы әр тестті келесі форматта оқиды:

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

Мұндағы, әр U_i ($1 \leq i < N$) өлшемі i болатын жиын, қай жолдар бар екенің сипаттайтын жиын. Кез келген $1 \leq i < N$ және $0 \leq j < i$ болатын i және j үшін:

- егер j және i жолмен байланысқан болса, онда $U_i[j]$ 1 болады;
- егер j және i байланыспаған болса, онда $U_i[j]$ 0 болады.

In each scenario, before calling `longest_trip`, the sample grader checks whether the density of the road network is at least D . If this condition is not met, it prints the message `Insufficient Density` and terminates.

If the sample grader detects a protocol violation, the output of the sample grader is `Protocol Violation: <MSG>`, where `<MSG>` is one of the following error messages:

- `invalid array`: in a call to `are_connected`, at least one of arrays A and B
 - is empty, or
 - contains an element that is not an integer between 0 and $N - 1$, inclusive, or
 - contains the same element at least twice.
- `non-disjoint arrays`: in a call to `are_connected`, arrays A and B are not disjoint.
- `too many calls`: the number of calls made to `are_connected` exceeds 32 640 over the current invocation of `longest_trip`, or exceeds 150 000 in total.
- `too many elements`: the total number of landmarks passed to `are_connected` over all calls exceeds 1 500 000.

Otherwise, let the elements of the array returned by `longest_trip` in a scenario be $t[0], t[1], \dots, t[l-1]$ for some nonnegative l . The sample grader prints three lines for this scenario in the following format:

- line 1: l
- line 2: $t[0] \ t[1] \ \dots \ t[l-1]$
- line 3: the number of calls to `are_connected` over this scenario

Finally, the sample grader prints:

- line $1 + 3 \cdot C$: the maximum number of calls to `are_connected` over all calls to `longest_trip`