



Longest Trip

IOI 2023-ի կազմակերպիչները մեծ անախորժության մեջ են: Նրանք մոռացել են պլանավորել Օպուստասերի եքսկուրսիան հաջորդ օրվա համար, բայց դեռ շատ ուշ չէ ...

Օպուստասերում կա N հատ տեսարժան վայր համարակալված 0-ից $N - 1$ թվերով: Տեսարժան վայրերի որոշ զույգեր միացված են *երկկողմանի ճանապարհներով*: Տեսարժան վայրերի ցանկացած զույկ միացված է ամենաշատը մեկ ճանապարհով: Կազմակերպիչները *չգիտեն*, որ տեսարժան վայրերն են միացված ճանապարհով:

Կասենք, որ Օպուստասերի գրաֆի *խտությունը ամենաքիչը* δ է, եթե ցանկացած 3 տարբեր տեսարժան վայրերի համար գոյություն ունի ամենաքիչը δ ճանապարհ նրանց միջև: Այլ կերպ ասած, ցանկացած (u, v, w) եռյակի համար, այնպիսին, որ $0 \leq u < v < w < N$, (u, v) , (v, w) և (u, w) ճանապարհներից ամենաքիչը δ հատը առկա են:

Կազմակերպիչները *գիտեն* D թիվը, այնպիսին որ գրաֆի *խտությունը ամենաքիչը* D է: Նկատեք, որ D թիվը 3-ից մեծ չի կարող լինել:

Կազմակերպիչները կարող են *հարցումներ* տալ Օպուստասերի հեռախոսավարուհուն գրաֆի մասին տեղեկություններ ձեռք բերելու համար: Ամեն հարցում իրենից ներկայացնում է տեսարժան վայրերի երկու զանգված՝ $[A[0], \dots, A[P - 1]]$ և $[B[0], \dots, B[R - 1]]$: Հարցման բոլոր տեսարժան վայրերը պետք է տարբեր լինեն, այսինքն.

- $A[i] \neq A[j] \ 0 \leq i < j < P$;
- $B[i] \neq B[j] \ 0 \leq i < j < R$;
- $A[i] \neq B[j] \ 0 \leq i < P \ և \ 0 \leq j < R$.

Հեռախոսավարուհին կտեղեկացնի, արդյոք գոյություն ունի A և B բազմությունների տեսարժան վայրերի միջև որևէ ճանապարհ: Ավելի կոնկրետ, հեռախոսավարուհին վերադարձնում է `true`, եթե գոյություն ունեն i և j թվեր, այնպես, որ $0 \leq i < P$ և $0 \leq j < R$, և $A[i]$ և $B[j]$ տեսարժան վայրերը միացված են ճանապարհով: Հակառակ դեպքում, հեռախոսավարուհին կվերադարձնի `false`:

Ուղևորությունը տարբեր *տեսարժան* վայրերի l երկարության հաջորդականություն է՝ $t[0], t[1], \dots, t[l - 1]$, որտեղ 0-ից $l - 2$ ցանկացած i -ի համար, $t[i]$ և $t[i + 1]$ տեսարժան

վայրերը միացված են ճանապարհով: l երկարության ուղևորությունը կանվանենք **ամենաերկար ուղևորություն** եթե գոյություն չունի $l + 1$ երկարության ուղևորություն:

Ձեր խնդիրն է գտնել որևէ ամենաերկար ուղևորություն հեռախոսավարուհուն հարցումներ տալու միջոցով: Խնդիրը մուլտիթեստային է:

Իրականացման մանրամասներ

Դուք պետք է ծրագրավորել հետևյալ ֆունկցիան.

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

- N . տեսարժան վայրերի քանակը:
- D . գրաֆի երաշխավորված խտությունը:
- Ֆունկցիան պետք է վերադարձնի $t = [t[0], t[1], \dots, t[l - 1]]$ հաջորդականություն, որը նկարագրում է ամենաերկար ուղևորությունը:
- Այս ֆունկցիան մեկ թեստում կարող է կանչվել **մի քանի անգամ**:

Այս ֆունկցիան կարող է կանչել հետևյալ ֆունկցիային.

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

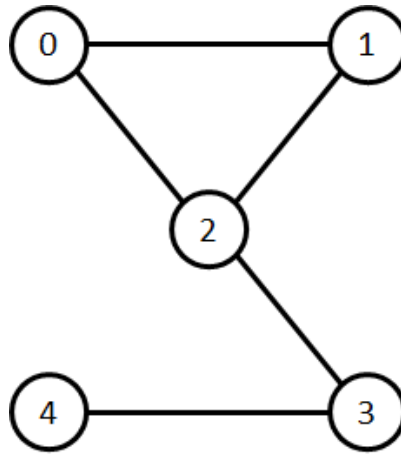
- A . տեսարժան վայրերի ոչ դատարկ զանգված:
- B . տեսարժան վայրերի ոչ դատարկ զանգված:
- A և B զանգվածները պետք է հատում չունենան:
- Ֆունկցիան կվերադարձնի `true`, եթե գոյություն ունի A -ից և B -ից քաղաքների զույգ, որոնք միացված են ճանապարհով: Այլապես այն կվերադարձնի `false`:
- Այս ֆունկցիան կարող է կանչվել ամենաշատը 32 640 անգամ, `longest_trip` ֆունկցիայի ամեն կանչի ժամանակ, և 150 000 անգամ ընդհանուր:
- A և B զանգվածների երկարությունների ընդհանուր գումարը `longest_trip` ֆունկցիայի կանչերի ժամանակ պետք է չգերազանցի 1 500 000-ը:

Գրեյդերը **ադապտիվ չէ** N և D թվերը, ինչպես նաև գրաֆը ֆիքսված են `longest_trip` ֆունկցիայի կանչից առաջ:

Օրինակներ

Օրինակ 1

Դիտարկենք դեպք, որտեղ $N = 5$, $D = 1$, և գրաֆը հետևյալն է.



`longest_trip` ֆունկցիան կանչվում է հետևյալ ձևով.

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

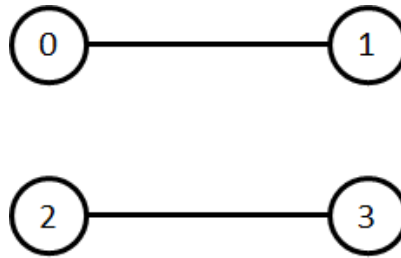
Այն կարող է օգտագործել `are_connected` ֆունկցիան.

Կանչ	Ճանապարհով միացված գույկերը	Վերադարձված արժեքը
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) և (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

Չորրորդ կանչից հետո պարզվում է, որ (1,4), (0,4), (1,3) և (0,3) գույգերից ոչ մեկը միացված է ճանապարհով: Քանի որ $D = 1$ (խտությունը), կարող ենք տեսնել, որ (0,3,4) եռյակի մեջ (3,4)-ին համապատասխան կողը պետք է առկա լինի: Նույն ձևով, 0 և 1 տեսարժան վայրերը պետք է միացված լինեն:

Այս պահին արդեն պարզ է, որ $t = [1, 0, 2, 3, 4]$ -ը 5 երկարության ուղևորություն է, և գոյություն չունի ավելի երկար ուղևորություն: Այսպիսով, `longest_trip` ֆունկցիան կարող է վերադարձնել `[1, 0, 2, 3, 4]`:

Դիտարկենք մեկ այլ դեպք, որտեղ $N = 4$, $D = 1$, և գրաֆը հետևյալն է.



`longest_trip` ֆունկցիան կանչվում է հետևյալ ձևով.

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

Այս դեպքում ամենաերկար ուղևորության երկարությունը 2 է: Այսպիսով, ինչոր `are_connected`-ի կանչերից հետո, `longest_trip` ֆունկցիան կարող է վերադարձնել $[0, 1]$, $[1, 0]$, $[2, 3]$ և $[3, 2]$ հաջորդականություններից որևէ մեկը:

Օրինակ 2

0 ենթախնդիրը պարունակում է $N = 256$ տեսարժան վայերով ևս մեկ օրինակ: Այս թեստը ներառված է `attachment package`-ում, որը կարելի է ներբեռնել մրցույթի համակարգից:

Սահմանափակումներ

- $3 \leq N \leq 256$
- N -երի գումարը `longest_trip`-ի կանչերում չի գերազանցում 1 024-ը:
- $1 \leq D \leq 3$

Ենթախնդիրներ

1. (5 միավոր) $D = 3$
2. (10 միավոր) $D = 2$
3. (25 միավոր) $D = 1$: Նշանակենք l^* -ով ամենաերկար ուղևորության երկարությունը: `longest_trip` ֆունկցիան պարտադրի չէ, որ վերադարձնի l^* երկարության ուղևորություն: Դրա փոխարեն, այն կարող է վերադարձնել ամենափչը $\left\lceil \frac{l^*}{2} \right\rceil$ երկարության հաջորդականություն:
4. (60 միավոր) $D = 1$

Եթե, որևէ թեստում `are_connected`-ի կանչը չի համապատասխանում վերոնշյալ պայմաններին, կամ `longest_trip`-ի վերադարձրած հաջորդականությունը ճիշտ չէ համապատասխան ենթախնդրի համար ձեր միավորը կլինի 0:

4-րդ ենթախնդրում Ձեր միավորը կախված է `are_connected`-ի կանչերի քանակից (`longest_trip`-ի մեկ կանչի ընդացքում): Թող q -ն լինի `longest_trip`-ի կանչերում

are_connected-ի կանչերի մաքսիմալ քանակը: Ձեր միավորը ենթախնդրի համար որոշվում է հետևյալ աղյուսակի միջոցով.

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

Գրեյդերի նմուշ

Դիցուք C -ն ցույց է տալիս դեպքերի քանակը, այսինքն longest_trip ֆունկցիայի կանչերի քանակը: Գրեյդերի նմուշը մուտքային տվյալները կարդում է հետևյալ ձևաչափով.

- տող 1: C

Ապա հաջորդում են C դեպքերի նկարագրությունները:

Գրեյդերի նմուշը կարդում է յուրաքանչյուր դեպքի նկարագրությունը հետևյալ ձևաչափով.

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

Այստեղ յուրաքանչյուր U_i ($1 \leq i < N$) դա i չափի զանգված է, որը նկարագրում է, թե որ վայրերն են միացված ճանապարհով: Յուրաքանչյուր i և j համար, այնպիսիք, որ $1 \leq i < N$ և $0 \leq j < i$,

- եթե j և i տեսարժան վայրերը միացված են ճանապարհով, ապա $U_i[j]$ -ի արժեքը կլինի 1,
- եթե j և i տեսարժան վայրերը միացնող ճանապարհ չկա, ապա $U_i[j]$ -ի արժեքը կլինի 0:

Յուրաքանչյուր դեպքում, նախքան longest_trip ֆունկցիան կանչելը, գրեյդերի նմուշը ստուգում է, արդյոք ճանապարհների ցանցի խտությունն առնվազն D է: Եթե այդ պայմանը տեղի չունի, այն տպում է Insufficient Density և ավարտում է աշխատանքը:

Եթե գրեյդերի նմուշը հայտնաբերում է պրոտոկոլի խախտում, ապա արտածում է Protocol Violation: <MSG>, որտեղ <MSG>-ն սխալների վերաբերյալ հետևյալ հաղորդագրություններից մեկն է:

- `invalid array. are_connected`-ի կանչում A և B զանգվածներից առնվազն մեկը
 - դատարկ է, կամ
 - պարունակում է 0-ից $N - 1$, ներառյալ, տիրույթին չպատկանող տարր, կամ
 - պարունակում է միևնույն տարրը առնվազն երկու անգամ,
- `non-disjoint arrays. are_connected` ֆունկցիայի կանչում A և B զանգվածները հատում ունեն,
- `too many calls. are_connected` ֆունկցիայի կանչերի քանակը գերազանցում է 32 640-ը `longest_trip`-ի ընթացիկ կանչի մեջ, կամ գերազանցումը է 150 000-ը ընդհանուրում:
- `too many elements. are_connected` ֆունկցիայի կանչին տրված տեսարժան վայրերի քանակը գերազանցում է 1 500 000-ը:

Հակառակ դեպքում, դիցուք դեպքում `longest_trip`-ի վերադարձրած զանգվածի տարրերն են $t[0], t[1], \dots, t[l - 1]$, որտեղ l -ն ոչ բացասական ամբողջ է: Այս սցենարի համար գրեյվերի նմուշը տպում է երեք տող հետևյալ ձևաչափով.

- տող 1: l
- տող 2: $t[0] \ t[1] \ \dots \ t[l - 1]$
- տող 3: սցենարում `are_connected` ֆունկցիայի կանչերի քանակը:

Վերջում գրեյվերի նմուշն արտածում է.

- տող $1 + 3 \cdot C$. `longest_trip`-ի բոլոր կանչերում `are_connected`-ի կանչերի մաքսիմում քանակը