



Մայրուղիների սակագներ (Highway Tolls)

Ճապոնիայում քաղաքները միացված են մայրուղիների ցանցով: Այդ ցանցը քաղկացած է N քաղաքից և M մայրուղուց: Յուրաքանչյուր մայրուղի միացնում է երկու տարբեր քաղաք: Քաղաքների ոչ մի գույգ միացված չէ երկու մայրուղով: Քաղաքները համարակալված են 0-ից $N - 1$, իսկ մայրուղիները՝ 0-ից $M - 1$ թվերով: Բոլոր մայրուղիներով երթևեկությունը երկկողմանի է: Այս մայրուղիների միջոցով կարող եք ցանկացած քաղաքից հասել ցանկացած այլ քաղաք:

Յուրաքանչյուր մայրուղով մեքենա վարելու համար սակագին է սահմանված: Սակագինը կախված է մայրուղում **երթևեկության** վիճակից: Երթևեկությունը կամ **թեթև** է, կամ **ծանր**: Երբ երթևեկությունը թեթև է, սակագինը A իեն է (Ճապոնական տարադրամ): Երբ երթևեկությունը ծանր է, սակագինը B իեն է: Երաշխավորվում է, որ $A < B$: Նկատի ունեցեք, որ դուք գիտեք A և B արժեքները:

Դուք ունեք մեքենա, որը ստանալով բոլոր մայրուղիներում երթևեկության պայմանները, հաշվում է, թե որքան է կազմում մինիմալ գումարային սակագինը S քաղաքից T քաղաք ($S \neq T$) հասնելու համար:

Սակայն մեքենան դեռևս փորձնական է: S և T արժեքները ֆիքսված են (այսինքն, մեքենայի ծրագրի մեջ են գրված) և դուք չգիտեք դրանք: Դուք կուզենայիք պարզել S -ը և T -ն: Դրա համար դուք պետք է մեքանային տաք երթևեկության պայմանների մի քանի տարբերակներ և, օգտագործելով նրա տպած սակագների արժեքները գտնեք S -ը և T -ն: Քանի որ երթևեկության պայմանները նկարագրելը ծախսատար է, դուք չեք ցանկանում մեքենան շատ անգամ օգտագործել:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ենթածրագիրը.

```
find_pair(int N, int[] U, int[] V, int A, int B)
```

- N : քաղաքների քանակը:
- U և V : M երկարության զանգվածներ, որտեղ M -ը քաղաքները միացնող մայրուղիների քանակն է: Յուրաքանչյուր i -ի համար ($0 \leq i \leq M - 1$) i մայրուղին միացնում է $U[i]$ և $V[i]$ քաղաքները:
- A : մայրուղու սակագինը, երբ երթևեկությունը թեթև է:
- B : մայրուղու սակագինը, երբ երթևեկությունը ծանր է:
- Այս ենթածրագիրը կանչվում է յուրաքանչյուր թեստի համար ճիշտ մեկ անգամ:

- Նկատի ունեցեք, որ M -ը զանգվածների երկարությունն է, և այն ստանալու եղանակը նշված է notice-ում:

`find_pair` ենթածրագիրը կարող է կանչել հետևյալ ֆունկցիան.

```
int64 ask(int[] w)
```

- w -ի երկարությունը պետք է լինի M : w զանգվածը նկարագրում է երթևեկության պայմանները:
- Յուրաքանչյուր i -ի ($0 \leq i \leq M - 1$) համար $w[i]$ -ն ցույց է տալիս երթևեկության վիճակը i -րդ մայրուղում: $w[i]$ -ի արժեքը պետք է լինի 0 կամ 1:
 - $w[i] = 0$ նշանակում է i -րդ մայրուղում երթևեկությունը թերթ է:
 - $w[i] = 1$ նշանակում է i -րդ մայրուղում երթևեկությունը ծանր է:
- Այս ֆունկցիան վերադարձնում է մինիմալ գումարային սակագիրը S -ից T հասնելու համար w -ում նշված երթևեկության պայմանների դեպքում:
- Այս ֆունկցիան կարող է կանչվել առավելագույնը 100 անգամ (յուրաքանչյուր թեստի համար):

`find_pair`-ը պատասխանը տալու համար պետք է կանչի հետևյալ ֆունկցիան.

```
answer(int s, int t)
```

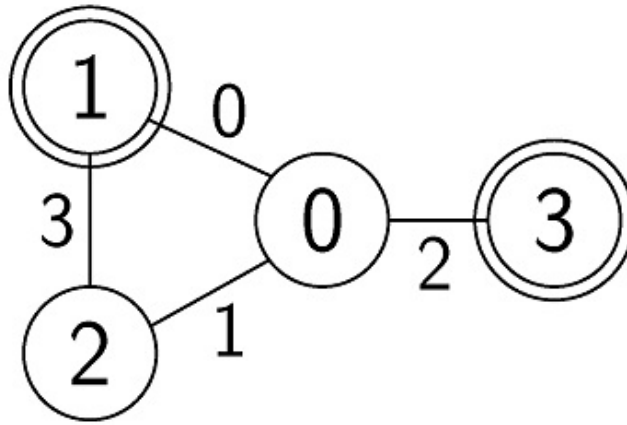
- s -ը և t -ն պետք է լինեն փնտրվող S և T թվագույգը (հերթականությունը կարևոր չէ):
- Այս ֆունկցիան պետք է կանչվի ճիշտ մեկ անգամ:

Եթե վերը նշված պայմաններից որևէ մեկը չբավարարվի, ձեր ծրագիրը կստանա **Wrong Answer**: Հակառակ դեպքում ձեր ծրագիրը կստանա **Accepted**, իսկ ձեր միավորը կախված է `ask`-ի կանչերի քանակից (տե՛ս ենթախնդիրները):

Օրինակ

Դիցուք $N = 4$, $M = 4$, $U = [0, 0, 0, 1]$, $V = [1, 2, 3, 2]$, $A = 1$, $B = 3$, $S = 1$ և $T = 3$:

Գրեյդերը կանչում է `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`:



Վերևի նկարում կողերի վրա նշված են մայրուղիների համարները: Ստորև բերված են ask-ի մի քանի հնարավոր կանչեր և համապատասխան վերադարձի արժեքները:

Կանչ	Վերադարձի արժեք
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

ask([0, 0, 0, 0]) ֆունկցիայի կանչի համար բոլոր մայրուղիների երթևեկությունը թեթև է, և բոլորի սակագինը 1 է: $S = 1$ -ից $T = 3$ հասնելու համար ամենաեժան ճանապարհը կլինի $1 \rightarrow 0 \rightarrow 3$: Այս ճանապարհի ընդհանուր սակագինը 2 է: Հետևաբար այս ֆունկցիան վերադարձնում է 2:

Ճիշտ պատասխան տալու համար find_pair ենթածրագիրը պետք է կանչի answer(1, 3) կամ answer(3, 1):

Կից արհիվացված փաթեթում sample-01-in.txt ֆայլը համապատասխանում է այս օրինակին: Փաթեթում առկա են այլ մուտքային տվյալներ ևս:

Սահմանափակումներ

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- For each $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ and $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)

- Դուք ցանկացած քաղաքից մայրուղիների միջոցով կարող եք հասնել ցանկացած այլ քաղաք:
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Այս խնդրում գրեյդերն ադապտիվ ՉԷ: Դա նշանակում է, որ S -ը և T -ն սկզբում ֆիքսված են և գրեյդերի աշխատանքի ընթացքում կախված չեն ձեր լուծման հարցումներից:

Ենթախնդիրներ

1. (5 միավոր) S -ից և T -ից մեկը 0 է, $N \leq 100$, $M = N - 1$
2. (7 միավոր) S -ից և T -ից մեկը 0 է, $M = N - 1$
3. (6 միավոր) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 միավոր) $M = N - 1$
5. (18 միավոր) $A = 1$, $B = 2$
6. (31 միավոր) Լրացուցիչ սահմանափակում չկա:

Ենթադրենք ձեր ծրագիրը ստացել է **Accepted**, և կատարել է ask -ի X կանչ: Այդ դեպքում ձեր թեստի համար P միավորը՝ կախված ենթախնդրի համարից, հաշվարկվում է հետևյալ կերպ.

- Ենթախնդիր 1. $P = 5$:
- Ենթախնդիր 2. If $X \leq 60$, $P = 7$: Հակառակ դեպքում $P = 0$:
- Ենթախնդիր 3. If $X \leq 60$, $P = 6$: Հակառակ դեպքում $P = 0$:
- Ենթախնդիր 4. If $X \leq 60$, $P = 33$: Հակառակ դեպքում $P = 0$:
- Ենթախնդիր 5. If $X \leq 52$, $P = 18$: Հակառակ դեպքում $P = 0$:
- Ենթախնդիր 6.
 - Եթե $X \leq 50$, $P = 31$:
 - Եթե $51 \leq X \leq 52$, $P = 21$:
 - Եթե $53 \leq X$, $P = 0$:

Ենթախնդրի համար տրվում է այդ ենթախնդրի համար պատրաստված թեստերի համար տրված միավորներից նվազագույնը:

Գրեյդերի օրինակ

Գրեյդերի օրինակը մուտքային տվյալները կարողում է հետևյալ ձևաչափով.

- տող 1: $N M A B S T$
- տող $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

Եթե ձեր ծրագիրը ստանում է **Accepted**, գրեյդերի օրինակը տպում է Accepted: q , որտեղ q -ն ask -ի կանչերի քանակն է:

- Եթե ձեր ծրագիրը ստանում է **Wrong Answer**, այն տպում է `Wrong Answer: MSG`, որտեղ `MSG`-ը կարող է լինել.
- `answered not exactly once`: answer ենթածրագրի կանչերի քանակը մեկ չէ:
- `w is invalid`: ask-ին տրված w-ի երկարությունը M չէ, կամ `w[i]`-ն ո՛չ 0 է, ո՛չ 1 որևէ i -ի համար ($0 \leq i \leq M - 1$):
- `more than 100 calls to ask`: ask ֆունկցիան կանչվել է ավելի քան 100 անգամ:
- `{s, t} is wrong`: answer ենթածրագիրը կանչվել է ոչ ճիշտ `s` և `t` գույգով: