

B. Swapping Cities

Time limit	2 s
Memory limit	512 MB

Նկարագրություն

Ինդոնեզիայում կա N քաղաք՝ համարակալված 0 -ից $N - 1$ թվերով: Կա նաև M երկկողմանի ճանապարհ՝ համարակալված 0 -ից $M - 1$ թվերով: Յուրաքանչյուր ճանապարհ երկու տարբեր քաղաքներ է միացնում: i -րդ ճանապարհը $U[i]$ -րդ քաղաքը միացնում է $V[i]$ -րդ քաղաքին և այդ ճանապարհով անցնելուց մեքենան ծախսում է $W[i]$ միավոր գազ: Քաղաքները կապված են իրար այնպես, որ ցանկացած քաղաքից կարելի է հասնել ցանկացած քաղաք:

Հաջորդ Q օրերից յուրաքանչյուրին երկու քաղաքներ ցանկանալու են քաղաքական կապ ստեղծել: Մասնավորապես, j -րդ օրը $X[j]$ -րդ քաղաքը ցանկանալու է քաղաքական կապ ստեղծել $Y[j]$ -րդ քաղաքի հետ: Դրա համար $X[j]$ -րդ քաղաքը մեքենայով ներկայացուցիչ է ուղարկելու $Y[j]$ -րդ քաղաք: Նմանապես, $Y[j]$ -րդ քաղաքը նույնպես մեքենայով ներկայացուցիչ է ուղարկելու $X[j]$ -րդ քաղաք:

Երթևեկություն չձանրաբեռնելու համար երկու մեքենաները չպետք է հանդիպեն: Մասնավորապես, երկու մեքենաները չպետք է միաժամանակ լինեն միևնույն քաղաքում: Նաև, երկու մեքենաները չպետք է երթևեկեն միևնույն ժամանակ միևնույն փողոցի երկու հանդիպակաց ծայրերից: Բացի այդ, եթե մեքենան մտել է որևէ ճանապարհ, ապա գնում է մինչև վերջ (այլ կերպ ասած, չի թույլատրվում, որ մեքենաները ճանապարհի կեսին U -աձև շրջադարձ անեն): Սակայն թույլատրվում է միևնույն քաղաքում լինել մի քանի անգամ և միևնույն ճանապարհով անցնել մի քանի անգամ: Նաև, մեքենաները կարող են ցանկացած քաղաքում սպասել որքան ցանկանան:

Քանի որ մեքենայի գինը կախված է վառելիքի բաքի չափից, երկու քաղաքները ցանկանում են ընտրել այնպիսի երթուղիներ երկու մեքենաների համար, որ նրանց վառելիքի բաքերի ծավալների մաքսիմումը մինիմալ լինի: Գազի լցակայաններ, պրակտիկորեն անսահմանափակ քանակի գազով, կան բոլոր քաղաքներում, այնպես որ մեքենայի բաքի ծավալը պետք է այնքան լինի, որ կարողանա անցնել իր երթուղում մաքսիմում գազի ծախս պահանջող ճանապարհը:

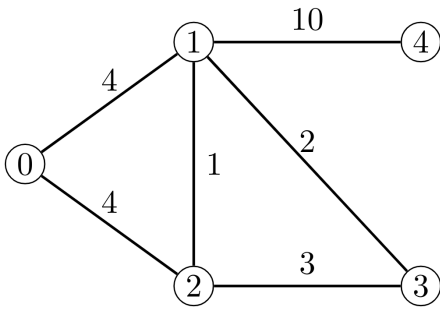
Խնդիրը

Դուք պետք է իրականացնեք `init` և `getMinimumFuelCapacity` ֆունկցիաները:

- `init(N, M, U, V, W)` - Այս ֆունկցիան գրեյդերի կողմից կանչվելու է միայն մեկ անգամ՝ նախքան `getMinimumFuelCapacity` ֆունկցիայի կանչը:
 - N : Ամբողջ թիվ, ցույց է տալիս քաղաքների քանակը:
 - M : Ամբողջ թիվ, ցույց է տալիս ճանապարհների քանակը:
 - U : M ամբողջ թվերի զանգված, ցույց են տալիս ճանապարհների առաջին ծայրակետերը:
 - V : M ամբողջ թվերի զանգված, ցույց են տալիս ճանապարհների երկրորդ ծայրակետերը:
 - W : M ամբողջ թվերի զանգված, ցույց են տալիս ճանապարհներով անցնելու համար գազի ծախսը:
- `getMinimumFuelCapacity(X, Y)` - Այս ֆունկցիան կանչվելու է գրեյդերի կողմից ճիշտ Q անգամ:
 - X : Ամբողջ թիվ, ներկայացնում է առաջին քաղաքը:
 - Y : Ամբողջ թիվ, ներկայացնում է երկրորդ քաղաքը:
 - Այս ֆունկցիան պետք է վերադարձնի մեկ ամբողջ թիվ, որը ցույց է երկու մեքենաներից մեծ գազի բաքով մեքենայի գազի բաքի մինիմալ ծավալը, ընդ որում այդ մեքենաներից մեկով X -րդ քաղաքի ներկայացուցիչը պետք է կարողանա գնալ Y -րդ քաղաք, իսկ մյուսով Y -րդ քաղաքի ներկայացուցիչը պետք է կարողանա գնալ X -րդ քաղաք հետևելով խնդրում նկարագրված կանոններին, կամ պետք է վերադարձնի -1 , եթե հնարավոր չէ դա անել:

Օրինակ

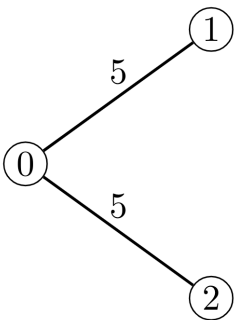
Առաջին օրինակում, $N = 5$, $M = 6$, $U = [0, 0, 1, 1, 1, 2]$, $V = [1, 2, 2, 3, 4, 3]$, $W = [4, 4, 1, 2, 10, 3]$, $Q = 3$, $X = [1, 2, 0]$, $Y = [2, 4, 1]$: Օրինակը պատկերված է հետևյալ նկարում.



Գրեյդերը սկզբում կանի `init(5, 6, [0, 0, 1, 1, 1, 2], [1, 2, 2, 3, 4, 3], [4, 4, 1, 2, 10, 3])` կանչը: Դրանից հետո գրեյդերը կանի հետևյալ կանչերը.

- `getMinimumFuelCapacity(1, 2)`: Նախ առաջին քաղաքից մեքենան կգնա երրորդ քաղաք: Հետո մյուս մեքենան երկրորդ քաղաքից կգնա առաջին քաղաք: Ապա երրորդ քաղաքից առաջին մեքենան կգնա երկրորդ քաղաք: Այսպիսով, մաքսիմալ վառելիքի ծախսը երկու մեքենաների համար 3 է (որն անհրաժեշտ է երրորդ քաղաքից երկրորդ քաղաք գնալու համար): Ավելի քիչ վառելիքի ծախսով երթուղիների ընտրություն չկա, հետևաբար ֆունկցիան պետք է վերադարձնի 3:
- `getMinimumFuelCapacity(2, 4)`: Չորրորդ քաղաք գնալ, գալու համար մեքենաները պետք է կարողանան տեղավորել 10 միավոր վառելիք: Հետևաբար այս դեպքում ֆունկցիան պետք է վերադարձնի 10:
- `getMinimumFuelCapacity(0, 1)`: Այս դեպքում ֆունկցիան պետք է վերադարձնի 4:

Երկրորդ օրինակում, $N = 3$, $M = 2$, $U = [0, 0]$, $V = [1, 2]$, $W = [5, 5]$, $Q = 1$, $X = [1]$, $Y = [2]$: Օրինակը պատկերված է հետևյալ նկարում.



Գրեյդերը սկզբում կանի `init(3, 2, [0, 0], [1, 2], [5, 5])` կանչը: Հետո կկատարի հետևյալ կանչը.

- `getMinimumFuelCapacity(1, 2)`: Առաջին քաղաքից երկրորդ քաղաք գնացող մեքենան չի կարող անել այնպես, որ չհանդիպի երկրորդ քաղաքից առաջին քաղաք գնացող մեքենային, հետևաբար ֆունկցիան պետք է վերադարձնի -1 :

Սահմանափակումներ

- $2 \leq N \leq 100\,000$.

- $N - 1 \leq M \leq 200\,000$.
- $0 \leq U[i] < V[i] < N$.
- Քաղաքների յուրաքանչյուր զույգի միջև կա առավելագույնը մեկ ճանապարհ:
- Ցանկացած քաղաքից ցանկացած քաղաք հնարավոր է գնալ:
- $1 \leq W[i] \leq 10^9$.
- $1 \leq Q \leq 200\,000$.
- $0 \leq X[j] < Y[j] < N$.

Ենթախնդիր 1 (6 միավոր)

- Յուրաքանչյուր քաղաք առավելագույնը երկու ճանապարհի ծայրակետ է:

Ենթախնդիր 2 (7 միավոր)

- $M = N - 1$.
- $U[i] = 0$.

Ենթախնդիր 3 (17 միավոր)

- $Q \leq 5$.
- $N \leq 1\,000$.
- $M \leq 2\,000$.

Ենթախնդիր 4 (20 միավոր)

- $Q \leq 5$.

Ենթախնդիր 5 (23 միավոր)

- $M = N - 1$.

Ենթախնդիր 6 (27 միավոր)

- Լրացուցիչ սահմանափակումներ չկան:

Գրեյդերի նմուշ

Գրեյդերի նմուշը կարդում է մուտքային տվյալները հետևյալ ձևաչափով.

```
N M
U[0] V[0] W[0]
U[1] V[1] W[1]
.
.
.
U[M-1] V[M-1] W[M-1]
Q
X[0] Y[0]
X[1] Y[1]
.
.
.
X[Q-1] Y[Q-1]
```

Յուրաքանչյուր `getMinimumFuelCapacity` կանչի համար գրեյդերի նմուշը տպում է այդ ֆունկցիայի վերադարձրած արժեքը: