

Task Collecting Diamonds

 3 sec.  4 MB

Ռոդոպի լեռներում հայտնաբերվել է ադամանդների հանքավայր: Պարզության համար համարենք, որ հանքավայրն ունի N սրահներ, որոնք համարակալված են 0 -ից $N - 1$ ամբողջ թվերով: Կան M միակողմանի միջանցքներ, որոնք կապում են որոշ սրահներ այնպես, որ յուրաքանչյուր սրահից դուրս եկող առնվազն մեկ միջանցք կա: Յուրաքանչյուր միջանցք ունի որոշակի ադամանդների քանակ, որոնք կարող են արդյունահանվել, երբ նրանով անցնում են: Այդ թիվը **չի փոփոխվում** երբ միջանցքով անցնում են, այն միշտ մնում է նույնը հետագա անցումների համար:

Հնարավոր է, որ միջանցքը կապի սրահն ինքն իր հետ, և կարող են լինել մեկից ավելի միջանցքներ միևնույն սրահների գույգի միջև (հնարավոր է նաև նույն ուղղությամբ): Նաև, չի երաշխավորվում, որ սրահները կապակցված են, այսինքն կարող է լինել սրահների (x, y) գույգ, որ y -ը հասանելի չլինի x -ից:

Պետարն անցնելու է K միջանցքներով ադամանդ արդյունահանելու համար: Նա ընտրելու է ինչ-որ s սրահ մեկնարկի համար, այնուհետև գնալու է ինչ-որ սրահ անցնելով s -ից սկսվող միջանցքով, և այդպես շարունակ այնքան, քանի դեռ նա չի անցել ճիշտ K միջանցքներ: Նկատի ունեցեք, որ նա կարող է կրկնել սրահներն ու միջանցքները, և որ միջանցքից հավաքվող ադամանդների քանակը չի փոխվում, երբ նա անցնում է այդ միջանցքով: Նկատեք, որ նա միշտ տարբերակ ունի անցնելու K միջանցքներով հաջորդաբար:

Պետարն ընտրելու է s -ը և ճանապարհը, որին հետևելու է, հետևյալ կերպ՝ սկզբից, նա ուզում է մաքսիմիզացնել առաջին միջանցքից հավաքվելիք ադամանդների քանակը: Բոլոր այդպիսի տարբերակներից նա ընտրելու է այն տարբերակը, որի դեպքում երկրորդ միջանցքից վերցրած ադամանդների քանակն է առավելագույնը: Սա կրկնվում է K անգամ: Այլ կերպ ասած, Պետարն ուզում է ընտրել լեքսիկոգրաֆիկորեն ամենամեծ ճանապարհը: Նրան հետաքրքրում է, թե ինչքան է լինելու իր հավաքած ադամանդների ընդհանուր քանակը, եթե նա ընտրի այդպիսի ճանապարհ: Օգնեք նրան հաշվել այն:



Implementation details

Դուք պետք է իրականացնեք `calculate_diamonds` ֆունկցիան:

```
long long int calculate_diamonds(int N, int M, int K,  
    std::vector<int> u, std::vector<int> v, std::vector<int> d)
```

- N : հանքավայրում սրահների քանակը,
- M : սրահների միջև միջանցքների քանակը,
- K : Պետարի անցնելիք միջանցքների քանակը,
- u, v, d : վեկտորներ բաղկացած M ամբողջ թվերից, որոնք ցույց են տալիս



միջանցքների սկզբնական սրահը, վերջնական սրահը, և աղամանդների քանակը:

Այս ֆունկցիան կանչվելու է մեկ անգամ յուրաքանչյուր թեստի համար և պետք է վերադարձնի մեկ թիվ՝ աղամանդների ընդհանուր թիվը, որը Պետարը կհավաքի օգտագործելով այդ ստրատեգիան:



Constraints

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$ for each $0 \leq i < M$
- Երաշխավորվում է, որ յուրաքանչյուր սրահի համար կա իրենից դուրս եկող առնվազն մեկ միջանցք:
- **Նկատեք անսովոր փոքր հիշողության սահմանափակումը՝ 4 MB:**



Subtasks

Ենթախնդ.	Միավոր	Պահանջվող ենթախնդ.	N	M	K	Լրացուցիչ սահմանափակումներ
0	0	—	—	—	—	Օրինակները:
1	11	0	≤ 10	≤ 20	≤ 10	—
2	10	0 — 1	≤ 100	$\leq 1\,000$	≤ 1000	—
3	26	0 — 2	≤ 100	$\leq 1\,000$	$\leq 10^9$	—
4	11	—	$\leq 2\,000$	$= N$	$\leq 10^9$	Յուրաքանչյուր սրահ ունի ճիշտ մեկ միջանցք, որն իրենից դուրս է գալիս, և ճիշտ մեկ միջանցք, որն ավարտվում է իրենում:
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Բոլոր $d[i]$ -ները տարբեր են:
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Կա ճիշտ մեկ $d[i] = 2$ ($0 \leq i < M$) d -ի բոլոր մյուս արժեքները 1 են:
7	21	0 — 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—

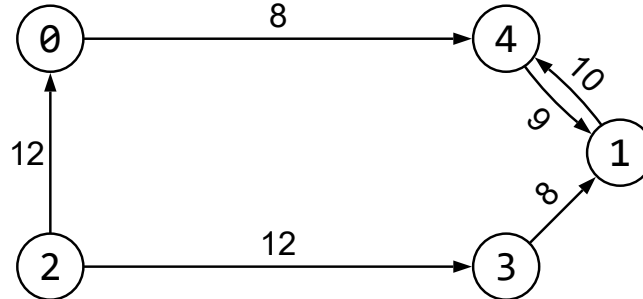


Ouplay 1

Դիտարկենք հետևյալ կանչն ու գծապատկերը, որտեղ $N = 5$, $M = 6$, և $K = 4$:



```
calculate_diamonds(5, 6, 4,
    {2, 0, 4, 2, 3, 1}, {0, 4, 1, 3, 1, 4}, {12, 8, 9, 12, 8, 10})
```



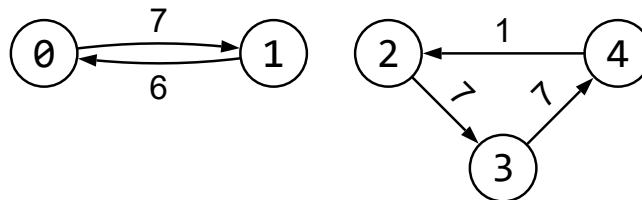
Պետարը կրնաորի անցնելու հետևյալ միջանցքները՝ $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$: Հավաքվող ադամանդների ընդհանուր քանակը կլինի 39, որը պետք է լինի ֆունկցիայի վերադարձվող արժեքը:



Օրինակ 2

Դիտարկենք հետևյալ կանչն ու գծապատկերը, որտեղ $N = 5$, $M = 5$, և $K = 4$:

```
calculate_diamonds(5, 5, 4,
    {0, 1, 2, 3, 4}, {1, 0, 3, 4, 2}, {7, 6, 7, 7, 1})
```



Կա 5 տարբերակ 4 միջանցք անցնելու համար՝

- (1) $0 \xrightarrow{7} 1 \xrightarrow{6} 0 \xrightarrow{7} 1 \xrightarrow{6} 0$;
- (2) $1 \xrightarrow{6} 0 \xrightarrow{7} 1 \xrightarrow{6} 0 \xrightarrow{7} 1$;
- (3) $2 \xrightarrow{7} 3 \xrightarrow{7} 4 \xrightarrow{1} 2 \xrightarrow{7} 3$;
- (4) $3 \xrightarrow{7} 4 \xrightarrow{1} 2 \xrightarrow{7} 3 \xrightarrow{7} 4$;
- (5) $4 \xrightarrow{1} 2 \xrightarrow{7} 3 \xrightarrow{7} 4 \xrightarrow{1} 2$.

(2)-րդ և (5)-րդ տարբերակները չեն մաքսիմիզացնում առաջին միջանցքից ստացվող ադամանդների քանակը: (1)-ին, (3)-րդ, և (4)-րդ տարբերակներից միայն (3)-րդն է մաքսիմիզացնում երկրորդ միջանցքից վերցված ադամանդների քանակը, հետևաբար սա Պետարի լավագույն ընտրությունն է: Նկատենք, որ (3)-րդ տարբերակը չի մաքսիմիզացնում երրորդ միջանցքից վերցված ադամանդների քանակը, չի մաքսիմիզացնում նաև ադամանդների ընդհանուր քանակը, սակայն դա միակ լեքսիկոգրաֆիկորեն ամենամեծ հաջորդականությունն է: Պետարի վերցրած ադամանդների ընդհանուր քանակը կլինի 22, որն էլ պետք է վերադարձվի ֆունկցիայի



կանչի դեպքում:



Գրեյդերի Նմուշը

Մուտքային տվյալների ձևաչափը այսպիսին է.

- տող 1. երեք ամբողջ թիվ - N , M , և K թվերի արժեքները:
- տող $1 + i$. երեք թիվ $u[i]$, $v[i]$, $d[i]$, որոնք նկարագրում են միջանցք, որը դուրս է գալիս $u[i]$ սրահից և վերջանում է $v[i]$ սրահում պարունակելով $d[i]$ ադամանդ արդյունահանման համա:

Ելքային տվյալների ձևաչափը այսպիսին է.

- տող 1. մեկ թիվ՝ ֆունկցիայի վերադարձվող արժեքը: