

Task Collecting Diamonds

 3 sec.  4 MB

Un zăcământ de diamant a fost descoperit în Munții Rodopi. Pentru simplitate, vom considera că zăcământul are N săli, numerotate cu numere întregi de la 0 la $N - 1$. Există M coridoare unidirecționale ce conectează unele săli, astfel încât există cel puțin un coridor care pleacă din fiecare sală. Fiecare coridor conține un anumit număr de diamante ce pot fi minate atunci când se trece prin el. Acest număr **nu se schimbă** atunci când trecem prin coridor de mai multe ori - rămâne mereu același și la parcurgerile ulterioare.

Este posibil ca un coridor să conecteze o sală cu ea însăși și se poate ca mai multe coridoare să conecteze aceeași pereche de săli (posibil în aceeași direcție). De asemenea, nu este garantat faptul că sălile sunt conectate între ele; cu alte cuvinte, ar putea exista o pereche de săli (x, y) astfel încât nu se poate ajunge în y plecând din x .

Petar va parcurge K coridoare pentru a mina diamante. El va alege o sală s din care va începe, apoi se va muta într-o sală mergând pe un coridor ce începe din s , și așa mai departe, până când va fi parcurs exact K coridoare. Observați faptul că el poate repeta săli sau coridoare, iar numărul de diamante pe care le colectează de pe un coridor nu se schimbă atunci când trece de mai multe ori prin el. Totodată, există întotdeauna o cale prin care el poate parcurge o secvență de K coridoare.

Petar va alege s și drumul pe care va merge în următorul mod: pentru început, el vrea să maximizeze numărul de diamante pe care le va colecta din primul coridor pe care îl parcurge. Dintre toate variantele acestea, el va alege să maximizeze numărul de diamante pe care le va colecta din al doilea coridor. Acest lucru se repetă de K ori. Cu alte cuvinte, Petar vrea să aleagă un drum maxim lexicografic. Acum el se întreabă care va fi numărul total de diamante pe care le va colecta dacă alege un astfel de drum. Ajuțați-l să calculeze acest lucru.



Detalii de implementare

Aveți de implementat funcția `calculate_diamonds`:

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

- N : numărul de săli din zăcământul de diamant;
- M : numărul de coridoare dintre săli;
- K : numărul de coridoare pe care Petar le va parcurge;
- u, v, d : vectori cu M numere întregi, ce reprezintă sala din care pleacă, sala în care ajunge, și numărul de diamante pentru fiecare coridor.

Această funcție va fi apelată o dată pentru fiecare test și trebuie să returneze un singur număr - numărul total de diamante pe care Petar le va colecta folosind strategia sa.



Restricții

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$ pentru orice $0 \leq i < M$
- Se garantează că există cel puțin un coridor care pleacă din fiecare sală
- **Observați limita de memorie de 4 MB.**



Subtask-uri

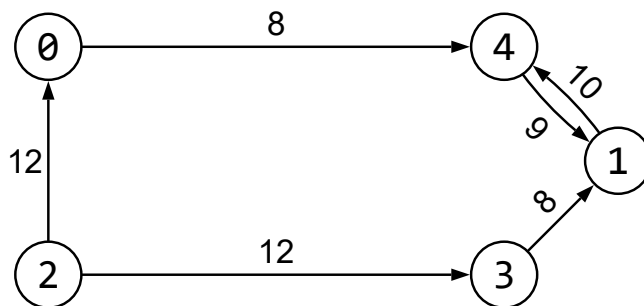
Subtask	Scor	Subtask-uri necesare	N	M	K	Restricții suplimentare
0	0	—	—	—	—	Exemple
1	11	0	≤ 10	≤ 20	≤ 10	—
2	10	0 – 1	≤ 100	$\leq 1\,000$	≤ 1000	—
3	26	0 – 2	≤ 100	$\leq 1\,000$	$\leq 10^9$	—
4	11	—	$\leq 2\,000$	$= N$	$\leq 10^9$	Fiecare sală are exact un coridor ce pleacă din ea și exact un coridor ce intră în ea
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Toate valorile $d[i]$ sunt distincte.
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Există exact o valoare $d[i] = 2$ ($0 \leq i < M$) și restul valorilor din d sunt egale cu 1.
7	21	0 – 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—



Exemplul 1

Considerăm apelul și imaginea de mai jos, în care $N = 5$, $M = 6$, și $K = 4$:

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

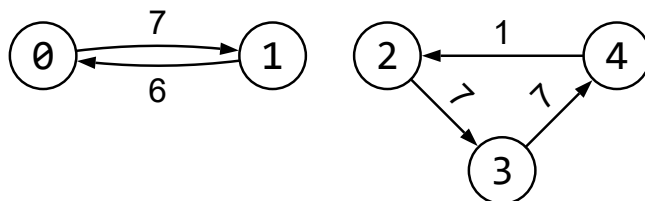


Petar va alege să parcurgă următoarele coridoare: $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$. Numărul total de diamante pe care le va colecta este 39, care ar trebui să fie valoarea returnată de apel.

Exemplul 2

Considerăm apelul și imaginea de mai jos, în care $N = 5$, $M = 5$, și $K = 4$:

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



Există 5 opțiuni de parcurgere a 4 coridoare:

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

Opțiunile (2) și (5) nu maximizează numărul de diamante colectate din primul coridor. Dintre opțiunile (1), (3) și (4) doar opțiunea (3) maximizează numărul de diamante din al doilea coridor, deci aceasta este cea mai bună opțiune pentru Petar. Observați că opțiunea (3) nu maximizează numărul de diamante din al treilea coridor, nici nu maximizează numărul total de diamante, dar este singura variantă maximă lexicografic. Numărul total de diamante pe care Petar le colectează este 22, care ar trebui să fie valoarea returnată de apel.

Exemplu de grader

Formatul datelor de intrare este următorul:

- linia 1: trei numere întregi - valorile N , M , and K .



- linia $1 + i$: trei numere întregi $u[i]$, $v[i]$, $d[i]$ - reprezentând un coridor ce pleacă din sala $u[i]$ și ajunge în sala $v[i]$ cu $d[i]$ diamante de colectat.

Formatul datelor de ieșire este următorul:

- linia 1: un număr întreg - valoarea returnată de apel.