



Zadatak Sakupljanje dijamantata

 3 sek.  4 MB

Otkriveno je nalazište dijamantata u planinama Rodop. Radi jednostavnosti, pretpostavićemo da nalazište ima N prostorija, označenih brojevima od 0 do $N - 1$. Postoji M jednosmjernih hodnika koji povezuju neke od prostorija na način da za svaku prostoriju postoji bar jedan hodnik koji izlazi iz nje. Svaki hodnik sadrži određeni broj dijamantata koji se mogu iskopati tokom prolaska. Taj broj **se ne mijenja** kada se prođe kroz hodnik - broj ostaje isti za svako od narednih prolazjenja.

Moguće je da je hodnik povezuje prostoriju samu sa sobom, i moguće je da postoji više hodnika koji povezuju jedan par prostorija (moguće u istom smjeru). Nije garantovano da su hodnici povezani, odnosno, može postojati par prostorija (x, y) takav da se do y ne može doći iz x .

Petar će proći kroz K hodnika da iskopa dijamante. Odabraće da počne sa nekom prostorijom s , zatim će preći u drugu prostoriju prolaskom kroz hodnik koji počinje u s , i nastaviće tako sve dok ne prođe kroz tačno K hodnika. Obratite pažnju da može ponovo prolaziti kroz iste hodnike i prostorije, i da se broj dijamantata koje prikupi iz hodnika ne mijenja prilikom ponovnog prolaska. Obratite pažnju da će uvijek biti način za njega da prođe kroz K uzastopnih hodnika.

Petar će odabrati s i put kojim će se kretati na naredni način: Prvo, nastojaće maksimizirati broj dijamantata koje prikupi kroz prvi hodnik kroz koji prođe. Među tim načinima, nastojaće maksimizirati broj dijamantata koje će prikupiti prolaskom kroz drugi hodnik. Ovaj postupak će ponoviti K puta. Drugim riječima, Petar želi odabrati leksikografski najveći put. Zanima ga koliko će ukupno dijamantata prikupiti ukoliko izabere ovakav put. Pomozite mu da to sračuna.



Detalji implementacije

Potrebno je da implementirate funkciju `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 : broj prostorija u nalazištu dijamantata;
- M : broj hodnika između prostorija;
- K : broj hodnika kroz koje će Petar proći;
- u, v, d : vektori M cijelih brojeva, koji predstavljaju početne prostorije, krajnje prostorije i broj dijamantata u hodnicima.

Ova funkcija će biti pozvana za svaki testni slučaj i treba da vrati jedan broj - ukupan broj dijamantata koje će Petar skupiti korištenjem ove strategije.



Ograničenja

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$ za svaki $0 \leq i < M$
- Garantovano je da za svaku prostoriju postoji bar jedan hodnik koji počinje iz nje.
- **Obratiti pažnju na neuobičajeno malo ograničenje na upotrebu memorije od 4 MB.**



Podzadaci

Podzadatak	Poeni	Zahtjevani podzadaci	N	M	K	Dodatna ograničenja
0	0	—	—	—	—	Primjeri.
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	—	$\leq 2\,000$	$= N$	$\leq 10^9$	Svaka prostorija ima tačno jedan hodnik koji počinje u njoj i tačno jedan hodnik koji završava u njoj.
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Sve vrijednosti $d[i]$ su jedinstvene.
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Postoji tačno jedan $d[i] = 2$ ($0 \leq i < M$) i sve ostale vrijednosti u d su jednake 1.
7	21	0 – 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—



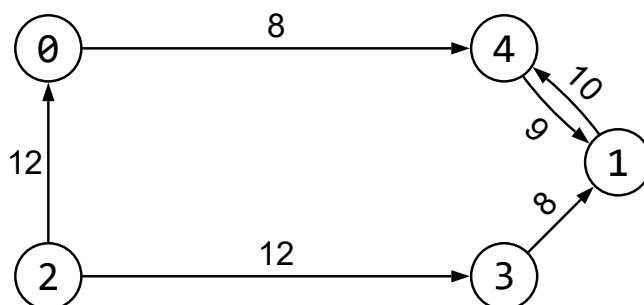
Primjer 1

Posmatrajmo naredni poziv i ilustraciju, za $N = 5$, $M = 6$, i $K = 4$:

```
calculate_diamonds(5, 6, 4,
```



$\{2, 0, 4, 2, 3, 1\}, \{0, 4, 1, 3, 1, 4\}, \{12, 8, 9, 12, 8, 10\}$



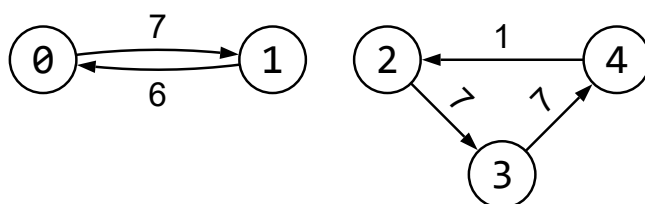
Petar će odabrati da prođe kroz naredne hodnike: $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$. Ukupan broj dijamantata koje će prikupiti je 39, što bi trebala biti vrijednost koju vraća pozvana funkcija.



Primjer 2

Posmatrajte naredni poziv i ilustraciju, za $N = 5$, $M = 5$, i $K = 4$:

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



Postoji 5 načina za prolazak kroz 4 hodnika:

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

Načini (2) i (5) ne daju maksimalan broj dijamantata iz prvog hodnika. Od načina (1), (3), i (4) jedino način (3) daje maksimalan broj dijamantata iz drugog hodnika tako da je ovo najbolji način za Petra. Obratiti pažnju da način (3) ne daje maksimalan broj dijamantata iz trećeg hodnika, niti daje maksimalan ukupan broj dijamantata, ali to je jedini leksikografski najveći niz. Ukupan broj dijamantata koje će Petar prikupiti je 22, što bi trebala biti vrijednost koju vraća pozvana funkcija.



Primjer ocjenjivača

Format ulaznih podataka je sljedeći:

- linija 1: tri cijela broja - vrijednosti N , M , i K .



- linija $1 + i$: tri cijela broja $u[i]$, $v[i]$, $d[i]$ – predstavljaju hodnik koji počinje u prostoriji $u[i]$ i završava u prostoriji $v[i]$ sa $d[i]$ dijamantata koji se mogu iskopati.

Format izlaznih podataka je sljedeći:

- linija 1: jedan cijeli broj – izlazna vrijednost pozvane funkcije.