



Задача Събиране на диаманти



3 sec.



4 MB

В Родопите е открито находище на диаманти. За простота ще приемем, че находището има N зали, обозначени с цели числа от 0 до $N - 1$. Има M еднопосочни коридори, свързващи някои от залите, така че има поне един коридор, излизащ от всяка зала. Всеки коридор има определен брой диаманти, които могат да бъдат добивани при преминаване през него. Това число **не се променя**, когато преминаваме през коридора – то остава същото при следващи преминавания. Възможно е един коридор да свързва една зала със себе си, а между една и съща двойка зали може да има множество коридори (възможно в една и съща посока). Също така не е гарантирано, че залите са свързани; т.е. може да има двойка зали (x, y) , такива че y да не може да бъде достигната от x .

Петър ще премине през K коридора, за да добива диаманти. Той ще избере някоя зала s за начало, след което ще се премести в зала, като премине през коридор, започващ от s , и така нататък, докато не премине през точно K коридора. Обърнете внимание, че той може да повтаря зали и коридори и че броят на диамантите, които събира от коридор, не се променя при повторение. Обърнете внимание, че винаги ще има начин той да премине през K коридора последователно.

Петър ще избере s и пътя, който ще следва до там, по следния начин: Първо, иска да увеличи максимално броя диаманти, които ще събере от първия коридор, през който преминава. Измежду всички такива опции, той ще увеличи максимално броя диаманти, които ще събере от втория коридор. Това се повтаря K пъти. Т.е. Петър иска да избере лексикографски най-голям път. Той се чуди какъв е общият брой диаманти, които ще събере, ако избере такъв път. Помогнете му да изчисли това.



Детайли по имплементацията

Трябва да имплементирате функцията `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 : броят на залите в диамантеното находище;
- M : броят на коридорите между залите;
- K : броят на коридорите, през които Петър ще премине;
- u, v, d : вектори от M цели числа, представляващи началните зали, крайните зали и диамантите в коридорите.

Тази функция ще бъде извикана веднъж за всеки тест и трябва да върне едно число – общия брой диаманти, които Петър ще събере, използвайки своята стратегия.

**Ограничения**

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$ за всяко $0 \leq i < M$
- Гарантирано е, че от всяка зала започва поне един коридор.
- **Обърнете внимание на необичайно малкия лимит на паметта 4 MB.**

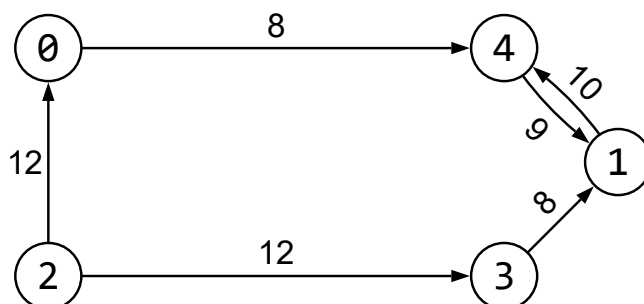
**Подзадачи**

Подзадача	Точки	Необходими подзадачи	N	M	K	Допълнителни ограничения
0	0	—	—	—	—	Примерите.
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	—	$\leq 2\,000$	$= N$	$\leq 10^9$	Всяка зала има точно един коридор, започващ от нея, и точно един коридор, завършващ в нея.
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Всички $d[i]$ са различни.
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	Има точно едно $d[i] = 2$ ($0 \leq i < M$) и всички останали стойности за d са равни на 1.
7	21	0 — 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—

**Пример 1**

Разгледайте следното извикване на функцията и илюстрацията за $N = 5$, $M = 6$, и $K = 4$:

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

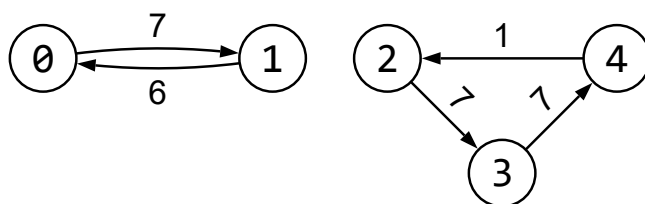


Петър ще избере да премине през следните коридори: $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$. Общият брой диаманти, които ще събере, е 39, която трябва да бъде стойността, върната от извикването.

**Пример 2**

Разгледайте следното извикване на функцията и илюстрацията за $N = 5$, $M = 5$, и $K = 4$:

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



Има 5 опции за преминаване през 4 коридора:

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

Опции (2) и (5) не максимизират броя на диамантите от първия коридор. От опции (1), (3) и (4) само опция (3) максимизира броя на диамантите от втория коридор, така че това е най-добрата опция за Петър. Обърнете внимание, че опция (3)



не максимизира броя на диамантите от третия коридор, нито максимизира общия брой диаманти, но е единствената лексикографски най-голяма последователност. Общият брой диаманти, които Петър ще събере, е 22, което би трябвало да е стойността, върната от извикването.



Примерен грейдър

Входният формат е следния:

- ред 1: три цели числа – стойностите на N , M , и K .
- ред $1 + i$: три цели числа $u[i]$, $v[i]$, $d[i]$ – представляващ коридор, започващ от зала $u[i]$ и завършващ в зала $v[i]$ с $d[i]$ диаманти за добив.

Изходният формат е следния:

- ред 1: едно цяло число – върнатата стойност при това извикване.