



აღმასების შეგროვება

⌚ 3 წმ 📄 4 მბ

როდოპის მთებში აღმასების საბადო აღმოაჩინეს. სიმარტივისათვის ვივარაუდოთ, რომ საბადოს აქვს N რაოდენობის დარბაზი, რომლებიც გადანომრილია მთელი რიცხვებით 0-დან $N-1$ -მდე. საბადოში არის M რაოდენობის ცალმხრივი დერეფანი, რომლებიც აერთებენ დარბაზებს ისე, რომ თითოეული დარბაზიდან გამოდის სულ მცირე ერთი დერეფანი მაინც. ყოველ დერეფანში არის გარკვეული რაოდენობის აღმასები, რომელთა მოპოვებაც შესაძლებელია მასში გავლისას. დერეფანში გავლისას ეს რიცხვი არ იცვლება - ის იგივე რჩება შემდგომი გავლისთვისაც.

შესაძლებელია, რომ დერეფანი დარბაზს საკუთარ თავთან აკავშირებდეს და დარბაზების ერთსა და იმავე წყვილს შორის (შესაძლოა, ერთი და იგივე მიმართულებით) რამდენიმე დერეფანი არსებობდეს. ასევე არ არის გარანტირებული, რომ ნებისმიერი ორი დარბაზი ერთმანეთთან დაკავშირებულია. ანუ, შეიძლება არსებობდეს დარბაზთა ისეთი (x, y) წყვილი, რომ x -დან y -მდე მიღწევა შეუძლებელი იყოს.

პეტარი აღმასების მოსაპოვებლად K რაოდენობის დერეფანს გაივლის. თავდაპირველად ის ირჩევს რომელიმე s დარბაზს, რომლიდანაც იწყებს. შემდეგ s -დან გამომავალი დერეფნის გავლით სხვა დარბაზში გადავა და ასე გააგრძელებს მანამ, სანამ ზუსტად K რაოდენობის დერეფანს არ გაივლის. გაითვალისწინეთ, რომ მას შეუძლია დარბაზებისა და დერეფნების განმეორებით გავლა და რომელიმე დერეფნის განმეორებით გავლისას ამ დერეფნიდან შეგროვებული აღმასების რაოდენობა არ იცვლება. შევნიშნოთ, რომ პეტარისათვის ყოველთვის შესაძლებელი იქნება K რაოდენობის დერეფნის თანმიმდევრობით გავლა.

პეტარი ირჩევს s -ს და გზას, რომელსაც გაივლის შემდეგნაირად: პირველ რიგში, მას სურს იმ აღმასების რაოდენობის მაქსიმიზაცია, რომელსაც შეაგროვებს პირველივე გავლილი დერეფნიდან. ყველაასეთ ვარიანტს შორის ის აირჩევს იმას, რომელიც შესაძლებელს გახდის მეორე დერეფნიდან შეგროვებული აღმასების მაქსიმიზაციას და ეს ყველაფერი K -ჯერ განმეორდება. სხვა სიტყვებით რომ ვთქვათ, პეტარს სურს ლექსიკოგრაფიულად უდიდესი გზის არჩევა. მას აინტერესებს, აღმასების რა ჯამურ რაოდენობას შეაგროვებს, თუ ასეთ გზას აირჩევს. დაეხმარეთ მას ამის გამოთვლაში.



იმპლემენტაციის დეტალები

თქვენ იმპლემენტაცია უნდა გაუკეთოთ შემდეგ ფუნქციას `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 : აღმასის საბადოში არსებული დარბაზების რაოდენობა;
- M : დარბაზებს შორის არსებული დერეფნების რაოდენობა;



- K : დერეფნების რაოდენობა, რომლებსაც პეტარი გაივლის;
- u, v, d : M ზომის მთელი რიცხვებისგან შემდგარი ვექტორები, რომლებიც წარმოადგენენ დერეფნის საწყის დარბაზებს, დერეფნის ბოლო დარბაზებს და დერეფნებში შეგროვებადი ალმასების რაოდენობას.

ეს ფუნქცია თითოეული ტესტისთვის ერთხელ გამოიძახება და უნდა დააბრუნოს ერთი რიცხვი - პეტარის მიერ თავისი სტრატეგიის გამოყენებით შეგროვებული ბრილიანტების საერთო რაოდენობა.



შეზღუდვები

- $1 \leq N \leq 2\,000$
- $1 \leq M \leq 4\,000$
- $1 \leq K \leq 10^9$
- $0 \leq u[i], v[i] < N$
- $1 \leq d[i] \leq 10^9$, როცა $0 \leq i < M$
- გარანტირებულია, რომ თითოეული დარბაზიდან ერთი დერეფანი მაინც იწყება.
- **შევნიშნოთ, რომ გვაქვს უზვეულოდ მცირე მეხსიერების შეზღუდვა - 4 MB.**



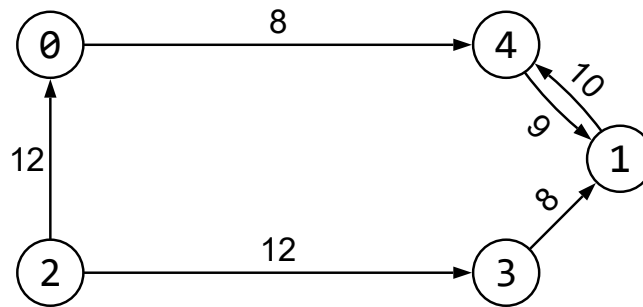
ქვეამოცანები

ქვეამოცანა	ქულა	საჭირო ქვეამოცანები	N	M	K	დამატებითი შეზღუდვები
0	0	—	—	—	—	მაგალითები.
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	—	$\leq 2\,000$	$= N$	$\leq 10^9$	თითოეული დარბაზისთვის არსებობს ზუსტად ერთ დერეფანი, რომელიც იწყება აქ და ზუსტად ერთი დერეფანი, რომელიც მთავრდება აქ.
5	10	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	ყველა $d[i]$ არის განსხვავებული.
6	11	—	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	არსებობს ზუსტად ერთი $d[i] = 2$ ($0 \leq i < M$) და ყველა სხვა მნიშვნელობა d -ში არის 1-ის ტოლი.
7	21	0 — 6	$\leq 2\,000$	$\leq 4\,000$	$\leq 10^9$	—

მაგალითი 1

განვიხილოთ შემდეგი გამოცხება და ილუსტრაცია $N = 5$, $M = 6$ და $K = 4$ -ისთვის:

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



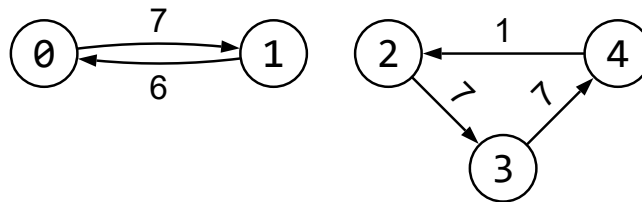
პეტარი აირჩევს შემდეგ კორიდორებში გავლას: $2 \xrightarrow{12} 3 \xrightarrow{8} 1 \xrightarrow{10} 4 \xrightarrow{9} 1$. მის მიერ შეგროვებული ბრილიანტების საერთო რაოდენობაა 39, რაც უნდა იყოს გამოძახების მიერ დაბრუნებული მნიშვნელობა.



მაგალითი 2

განიხილეთ შემდეგი გამოძახება და ილუსტრაცია, $N = 5$, $M = 5$ და $K = 4$ -ისთვის:

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



There are 5 options for passing through 4 corridors:

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

ვარიანტები (2) და (5) არ უკეთებენ მაქსიმიზაციას პირველი დერეფნიდან ბრილიანტების რაოდენობას. ვარიანტებიდან (1), (3) და (4) მხოლოდ ვარიანტი (3) უკეთებს მაქსიმიზაციას მეორე დერეფნიდან ბრილიანტების რაოდენობას, ამიტომ ეს საუკეთესო ვარიანტია პეტარისთვის. გაითვალისწინეთ, რომ ვარიანტი (3) არ უკეთებს მაქსიმიზაციას მესამე დერეფნიდან ბრილიანტების რაოდენობას და არც ბრილიანტების საერთო რაოდენობას, მაგრამ ეს არის ერთადერთი ლექსიკოგრაფიულად უდიდესი თანმიმდევრობა. პეტარის მიერ შეგროვებული ბრილიანტების საერთო რაოდენობაა 22, რაც უნდა იყოს გამოძახებით დაბრუნებული მნიშვნელობა.



Sample grader

The input format is the following:

- line 1: three integers – the values of N , M , and K .
- line $1 + i$: three integers $u[i]$, $v[i]$, $d[i]$ – representing a corridor starting from hall $u[i]$ and ending in hall $v[i]$ with $d[i]$ diamonds for mining.

The output format is the following:

- line 1: one integer – the return value of the call.



სანიმუშო გრეიდერი

შეყვანის ფორმატი შემდეგნაირია:

- ხაზი 1: სამი მთელი რიცხვი – N , M და K .
- ხაზი $1 + i$: სამი მთელი რიცხვი – $u[i]$, $v[i]$, $d[i]$ – წარმოადგენს კორიდორს, რომელიც იწყება დარბაზიდან $u[i]$, მთავრდება დარბაზში $v[i]$ და შესაძლებელია $d[i]$ ცალი აღმასის მოპოვება.

გამომავალი ფორმატი შემდეგნაირია:

- ხაზი 1: ერთი მთელი რიცხვი – გამოძახების დაბრუნებული მნიშვნელობა.