Minimal cost spanning tree

PODSECANJE:

U priority_queue se moze cuvati struktura podataka max-heap tako da uzimanjem elementa sa pocetka uvek dobijamo najveci element.

Vremenska složenost metode top() je i dalje O(1), dok O(log n) je vremenska složenost metode push(x) i pop().

Ako smestimo niz elemenata u priority_queue, i uzimamo elemente redom, onda smo uspeli sortirati niz u vremenu O(n log n). Želimo li raditi sa min hipom, možemo napraviti korisnički metod za poređenje (jednostavna zamena operatorom >).
Pri definisanju minHipa koristimo konstruktor koji zahteva 2 dodatna argumenta:

priority_queue<int,vector, poredi > minHip;
Dakle, u tom konstruktoru, osim tipa podataka koji cemo cuvati u minHipu, moramo navesti i nad kojim kontejnerom se izgradjuje minHip(najbolje staviti vector), kao i ime strukture ili klase koja ima preoptrecen operator() radi poredjenja

Primov algoritam

O(E*log V)

```
#include <bits/stdc++.h>
using namespace std;

const int N=1e5+10;

vector<pair<int, int> > g[N];
int n, m;
bool added[N];

long long Prim (int source)
{
   long long mst=0;
   priority_queue<pair<int, int>, vector<pair<int, int> >, greater<pair<int, int> > > pq;
   for (auto pxt: g[source]) pq.push(pxt);

   while (!pq.empty())
   {
      int v=pq.top().first;
      int w=pq.top().second;
      if (added[v]) continue;
      added[v]=1;
      mst+=1ll*w;

      for (auto pxt: g[v]) if (!added[pxt.second]) pq.push(pxt);
   }

   return mst;
```

```
}

int main()
{
   scanf ("%d%d", &n, &m);
   for (int i=0;i<m;i++)
   {
      int a, b, w;
      scanf ("%d%d%d", &a, &b, &w);
      g[a].push_back({w, b});
      g[b].push_back({w, a});
   }

   printf ("%lld", Prim(1));

   return 0;
}
```

Zadatak 1

U zemlji S, postoji *n* jezera (numerisanih od 1 do *n*) i *m* kanala između njih. Poznata je širina svakog kanala (u metrima). Kretanje kanalima se može izvesti u oba smera. Poznato je da čamac širine jedan metar može dospeti do ma kog jezera, počevši od jezera sa rednim brojem 1. Napisati program koji izračunava minimalni broj kanala koje treba proširiti, tako da čamac širine *k* metara može putovati između svaka dva jezera (čamac se može kretati od jednog jezera do drugog, ako je njegova širina manja ili jednaka od širine kanala koji povezuje jezera).

Ulaz

U prvoj liniji standardnog ulaza su dati celi brojevi *n* i *m* ($1 < n \leq 1000$, $1 < m \leq 100000$).

U narednih *m* linija su data tri cela broja, *i*, *j* i *w*, koji ukazuju da postoji kanal širine *w* ($1 \leq w \leq 200$) između jezera, *i* i *j* ($1 \leq i, j \leq n$). U poslednjoj liniji je dat ceo broj *k* ($1 \leq k \leq 200$).

Izlaz

U jedinom redu standardnog izlaza ispišite jedan ceo broj: minimalni broj kanala koji treba proširiti.

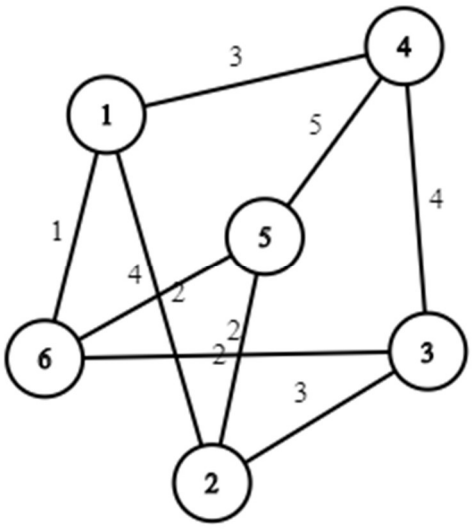
Primer

Ulaz

```
6 9
1 6 1
1 2 2
1 4 3
2 3 3
2 5 2
3 4 4
3 6 2
4 5 5
5 6 4
4
```

Izlaz

```
2
```



Rešenje;
1 način
Nađimo maksimalno obuhvatno stablo T_{max} grafa jezera *G* i povežimo ga kanalima. Posle izračunavamo koliko (grana) kanala iz T_{max} su uži od date vrednosti *K*. Neka je broj takvih kanala *q*. Ako se ovi kanali prošire, čamac širine *K* može da prođe između svaka dva jezera. Štaviše, ako uklonimo sve kanale uže od *K*, graf će se razbiti na *q+1* komponenti povezanosti i minimalni broj kanala koji će povezati sve komponente jeste *q*.
2 način
Tražimo broj komponenti povezanosti grafa jezera G_K i kanala širine barem *K.* Neka je taj broj *q+1*. Pošto je grad *G* povezan, postojaće *q* tesnih kanala, koji će pri proširivanju do širina *K* i dodavanjem u G_K povezati komponente. Pronalaženje komponenti povezanosti u G_K može da se obavi nekim algoritmom za obilazak grafa – *BFS* ili *DFS*.

Zadatak 2

U zemlji S, postoji *N* jezera (numerisanih od 1 do *n*) i *M* kanala između njih. Poznata je širina svakog kanala (u metrima). Kretanje kanalima se može izvesti u oba smera.

Transportno preduzeće *Jezero* je pripremilo listu od *K* parova jezera između kojih će se obavljati redovni transport robe i ljudi putem čamaca.
Napišite program koji izračunava maksimalnu širinu čamaca, koji mogu proći između parova jezera koji se nalaze na listi jezera (čamac se može kretati od jednog jezera do drugog, ako je njegova širina manja ili jednaka od širine kanala koji povezuje jezera).
Ulaz
U prvoj liniji standardnog ulaza su dati celi brojevi *N, M*, *K* ($N \leq 1000$, $M \leq 100000$, $K \leq 10000$).
U narednih *m* linija su data tri cela broja, *i*, *j* i *w*, koji ukazuju da postoji kanal širine *w* između jezera, *i* i *j* ($1 \leq i, j \leq N$, $Wi,j \leq 200$).

Potom sledi *K* linija, tako da svaka sadrži brojeve dva jezera *i*, *j*, između kojih se obavlja transport ljudi i robe.

Izlaz

Program treba da ispiše K redova na standardnom izlazu, od kojih svaki sadrži jedan ceo broj, jednak maksimalnoj širini čamca, koji može putovati između odgovarajuća dva jezera.

Primer

Ulaz

```
6 9 4
1 2 2
1 4 3
1 6 1
2 3 3
2 5 2
3 4 4
3 6 2
4 5 5
5 6 4
2 6
3 5
1 2
4 6
```

Izlaz

```
3
4
3
4
```

Rešenje:

Pronađimo maksimalno obuhvatno stablo T grafa jezera G i povežimo ga kanalima. Sve grane ovog stabla su grane sa najvećim težinama u G, odnosno najširi kanali među čvorovima grafa G. Prim-ovim algoritmom se može generisati ovo stablo. Možemo ovo stablo predstaviti kao koreno stablo tako da niz D čuva i reguliše širinu kanala (iz stabla) dok niz Pi čuva prethodnike čvorova. Uz malo analize obilaska stabla možemo izračunati na kom nivou u odnosu na koren je svaki čvor (vrednosti niza *Depth*). Maksimalno obuhvatno stablo koje je predstavljen na takav način omogućuje da se u vremenu $O(p)$ (gde p je dužina puta) izračuna širina puta između svakog para čvorova u datoj listi sa K čvorova.

Zadatak 3.

https://csacademy.com/contest/archive/task/x-distance/statement/

X Distance

Time limit: *1000 ms*

Memory limit: *128 MB*

You are given a weighted undirected graph with N nodes and M edges. We define the cost of a path to be equal to the maximum weight of an edge on the path. Find the number of pairs of nodes for which the minimum cost of a path between them is equal to X.