Minimalno povezujuće stablo

Ulaz

$G = (V, E)$ - neusmereni povezani težinski graf

Izlaz

Povezani podgraf koji sadrži sve čvorove takav da mu je suma cena grana minimalna.

Rešenje

Indukcijom po broju grana.

Baza: Rešenje mora da sadrži granu najmanje cene.

Induktivni korak: Podgraf T, broj grana $k < |V| - 1$.

Neka je E_k skup grana koje povezuju čvorove u T sa čvorovima van. Grana najmanje cene u E_k mora biti u rezultatu.

Algoritam

```
T = {};

for (w: cvorovi) {
    w.oznaka = false;
    w.cena = ∞;
}

(x, y) = grana sa najmanjom cenom u G;

x.oznaka = true;

for (grane (x, z)) {
    z.grana = (x, z);              // grana najmanje cene od T do z
    z.cena  = cenaGrane(z.grana); // cena grane
}

while (postoji neoznačen čvor) {
    w = neoznačen čvor najmanje vrednosti w.cena;

    if (w.cena = ∞) {
        greška: G nije povezan

    } else {
        w.oznaka = true;
        dodaj w.grana u T;

        for (sve grane (w, z)) {
            if (! z.oznaka) {
                if (cenaGrane(w, z) < z.cena) {
```

```
                    z.grana = (w, z);
                    z.cena = cenaGrane(z.grana);
                }
            }
        }
    }
}
```

Minimalno drvo razapinjanja

Posmatrajmo skup od *n* računara koje treba povezati optičkim kablovima. Poznata je cena povezivanja za svaka dva računara. Povezivanje treba izvršiti tako da ukupna cena bude minimalna. Taj problem je u teoriji grafova poznat kao problem formiranja minimalnog drveta razapinjanja.

Stablo koje dobijamo uklanjanjem određenog broja grana grafa, a da pritom dobijeni graf ostane povezan, zovemo razapinjajuće ili obuhvatno stablo (spanning tree). Generalno, konstruisanje razapinjajućeg stabla jednostavan je postupak, ali je često u primenama poželjno da takvo razapinjajuće stablo ima i neka dodatna svojstva, kao što je minimalna/maksimalna suma težina grana. Obuhvatna stabla se generišu algoritmima za obilazak grafa po širini ili dubini. Za isti graf može postojati više obuhvatnih stabala.

Minimalno obuhvatno stablo ima najmanju cenu (suma cena grana). Može biti više takvih stabala (ista cena).

PRIM I KRUSKAL ALGORITMI ZA KONSTRUKCIJU MCST
Neka je $G=(V,E)$ povezan neorijentisan graf i neka je $d : E \rightarrow R^+$ funkcija kojom su definisane dužine ivica. Minimalno drvo razapinjanja grafa G je podgraf $T=(V, E_T)$ gde je $E_T \subseteq E$ takav da važi:

- T je drvo (povezan graf bez ciklusa)
- zbir dužina ivica skupa E_T je minimalan.

Prikazujemo Primov algoritam za određivanje minimalnog drveta razapinjanja.

Drvo kreiramo tako što polazimo od proizvoljnog čvora (npr.0) pa dodajemo čvor po čvor, dok ne dodamo sve čvorove grafa. Razlikujemo skup U čvorova koje smo dodali u drvo i skup $V\setminus U$ čvorova koje još nismo dodali u drvo. Na početku je $U=\{0\}$ i $E_T=\{\}$. U svakoj iteraciji algoritma izaberemo granu (u,v) sa minimalnom dužinom takvu da je $u \in U$ i $v \in V\setminus U$, tj. biramo granu koja je najbliža kreiranom drvetu. Skup čvorova U proširimo za čvor v, $U=U\cup\{v\}$, a granu (u,v) dodamo u skup ivica traženog drveta $E_T= E_T\cup\{(u,v)\}$. Postupak ponavljamo dok ne postane $U=V$.

Pri realizaciji ovog algoritma, slično kao kod Dijkstrinog algoritma, koristimo

- niz *mark* kojim definišemo da li je čvor uključen u drvo ili nije

- niz *d* kojim pamtimo za svaki čvor $v \in V \setminus U$ (*mark[v]=false*) najmanju dužinu ivice koja spaja čvor *v* sa nekim čvorom *u* iz drveta (*mark[v]=true*)

- niz p kojim pamtimo za svaki čvor $v \in V \setminus U$ indeks njemu najbližeg čvora iz kreiranog drveta.

Rezultat algoritma, minimalno drvo razapinjanja definišemo granama koje čine to drvo. Početke grana pamtimo nizom *e1* a krajeve nizom *e2*.

Prim-ov algoritam:
PRIM(G, s)
$U = \{s\}$
$E' = \varnothing$
while ($U \neq V$) **do**
    find $(u, v) \Rightarrow \min \{w(u, v) : (u \in U) \text{ and } (v \in (V - U))\}$
    $U = U + \{v\}$
    $E' = E' + \{(u, v)\}$
end_while
$MCST = (U, E')$

Kruskal-ov algoritam
1. Inicijalno, graf se posmatra kao potpuno nepovezan (nepovezane komponente)
2. Skup grana E se uređuje po neopadajućoj težini (prioritetan red)
3. Nova grana se dodaje samo ako spaja dve odvojene komponente (T)
Kompleksnost Kruskalovog algoritama je O(m·log(n)), tj. vremenska slozenost je O(E log V)

KRUSKAL(G)
$E' = \varnothing$
for each $(u, v) \in E$ **do**
    PQ-INSERT(PQ, $w(u, v)$)
end_for
$num = 0$
while ($num < n - 1$) **do**
    $w(u,v)$ = PQ-MIN-DELETE(PQ)
    if (($u \in Ti$) and ($v \in Tj$) and ($i \neq j$)) **then**
        $E' = E' + \{(u, v)\}$
        $Tk = Ti + Tj$
        $num = num + 1$
    end_if
end_while
$MCST = (V, E')$

SLOŽENOST:
Koristeći binarni heap, Prim-Jarnikov algoritam je kompleksnosti O((m+n) ·log(n)) tj. slozenost je O((V+E)log V), a koristeći Fibonacci heap, kompleksnost algoritma je O(m+n·log(n)). Sa potpunim grafom, u prvom slučaju kompleksnost je $O(n^2 \cdot \log(n))$, a u drugom $O(n^2)$.

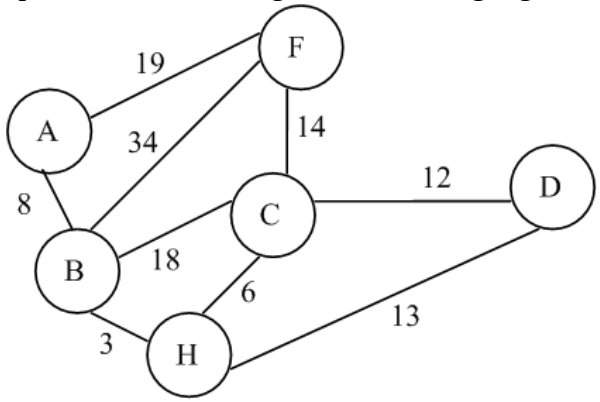
Poredjenje tri algoritma za konstrukciju MCST (Prim vs Kruskal vs Boruvka)
Osnovne razlike između tri algoritama su:

• Prim-Jarnikov algoritam u svakom koraku proširuje označeno stablo sa najbližim čvorom,
• Kruskalov algoritam u svakom koraku spaja dva najbliža stabla sa novom granom,

• Borůvkin algoritam u svakom koraku spaja sva najbliža stabla sa novom granom.

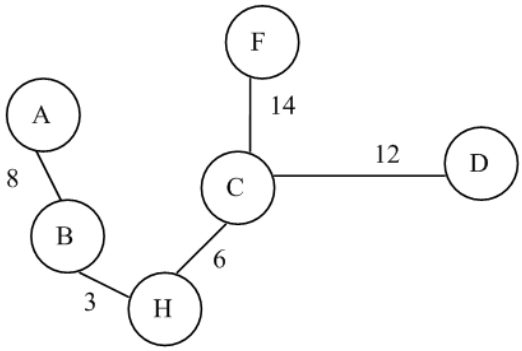
Kruskalov i Borůvkin algoritam se mogu unaprediti tako da njihova kompleksnost bude O(m·α(n)), gde je α inverzna Ackermanova funkcija.

Zadatak 1. Za dati graf na slici, konstruišite razapinjuće stablo minimalne cene (MCST) upotrebom Prim-ovog i Kruskal-ovog algoritma.



Rešenje:

PRIMov algoritam:
Čvor A je odabran za koren stabla



A-B
A-B, B-H
A-B, B-H, H-C
A-B, B-H, H-C, C-D
A-B, B-H, H-C, C-D, C-F
Cena MCST-a je 8+3+6+12+14=43

Prim-ov algoritam:
PRIM(G, s)
$U = \{s\}$
$E' = \varnothing$
while ($U \neq V$) **do**
    find $(u, v) \Rightarrow \min \{w(u, v) : (u \in U) \text{ and } (v \in (V - U))\}$
    $U = U + \{v\}$
    $E' = E' + \{(u, v)\}$
end_while
$MCST = (U, E')$

Kruskal-ov algoritam
1. Inicijalno, graf se posmatra kao potpuno nepovezan (nepovezane komponente)
2. Skup grana E se uređuje po neopadajućoj težini (prioritetan red)
3. Nova grana se dodaje samo ako spaja dve odvojene komponente (T)
Kompleksnost Kruskalovog algoritama je O(m·ln(n)).

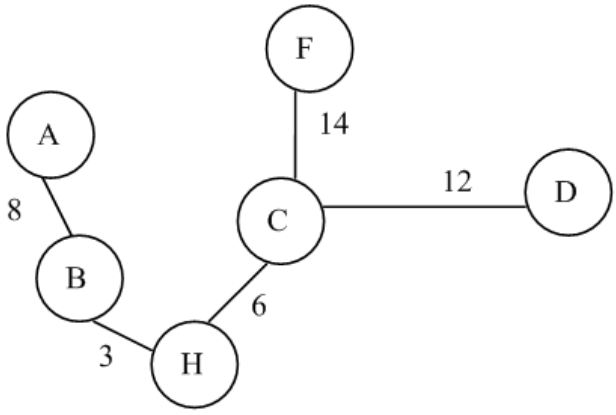
KRUSKAL(*G*)
E′ = ∅
for each (*u, v*) ∈ *E* **do**
PQ-INSERT(*PQ*, *w*(*u, v*))
end_for
num = 0
while (*num* < *n* - 1) **do**
w(*u,v*) = PQ-MIN-DELETE(*PQ*)
if ((*u* ∈ *Ti*) and (*v* ∈ *Tj*) and (*i* ≠ *j*)) **then**
E′ = *E′* + {(*u, v*)}
Tk = *Ti* + *Tj*
num = *num* + 1
end_if
end_while
MCST = (*V*, *E′*)

Sortirane grane {BH=3, CH=6, AB=8, CD=12, DH=13, CF=14, BC=18, AF=19, BF=34}

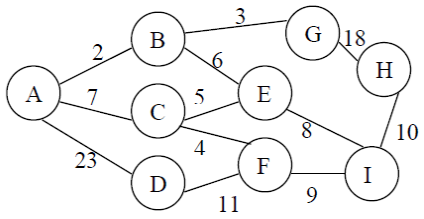
Kruskal – korak 1) Od korena stabla mora da polazi grana najmanje težine. Ovde se bira čvor H.

Kruskal – korak 2)
H-B
H-B, H-C
H-B, H-C, B-A
H-B, H-C, B-A, C-D
H-B, H-C, B-A, C-D, C-F

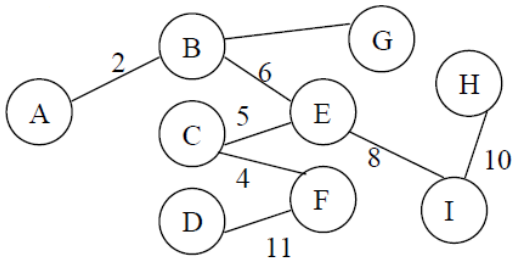
Kruskal – korak 3)



Zadatak 2.
Za dati graf na slici, konstruišite razapinjuće stablo minimalne cene (MCST) upotrebom Prim-ovog i Kruskal-ovog algoritma.



Rešenje:
Obuhvatno stablo



Prim-ov algoritam
A-B, B-G, B-E, E-C, C-F, E-I, I-H, F-D

Kruskal-ov algoritam
A-B, B-G, C-F, E-C, B-E, E-I, I-H, F-D

Zadatak 3

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.

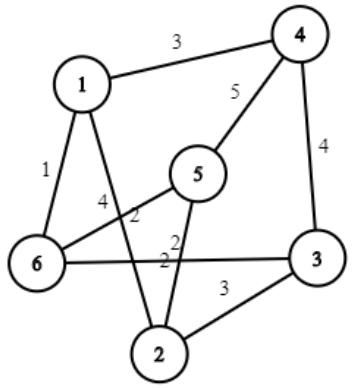
Primer

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 4

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.