Algoritmusok és adatszerkezetek gyakorlat – 11

Gráfalgoritmusok II.

Legrövidebb utak

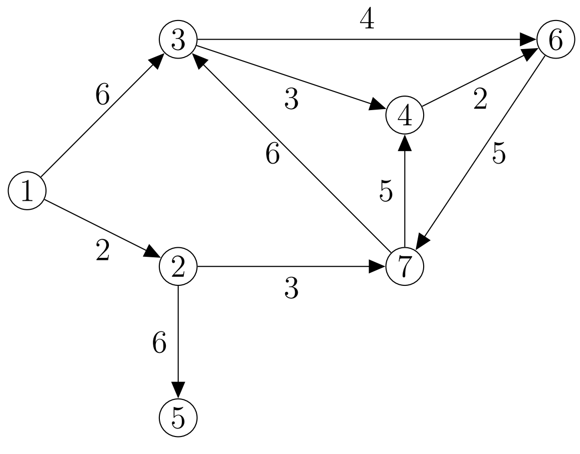
A feladat egy súlyozott gráfban egy adott pontból kiinduló legrövidebb utak megkeresése. Az input a G súlyozott gráf és a kiindulási s pont. Outputként egy legrövidebb utak fáját adunk vissza, egy Apa függvény által, továbbá a legrövidebb utak hosszait egy d függvény által. A feladatot nemnegatív élsúlyok esetén a következő Dijsktra algoritmussal oldhatjuk meg. A pontokat egy d érték szerinti Q módosítható prioritási sorban tároljuk.

Algoritmus:

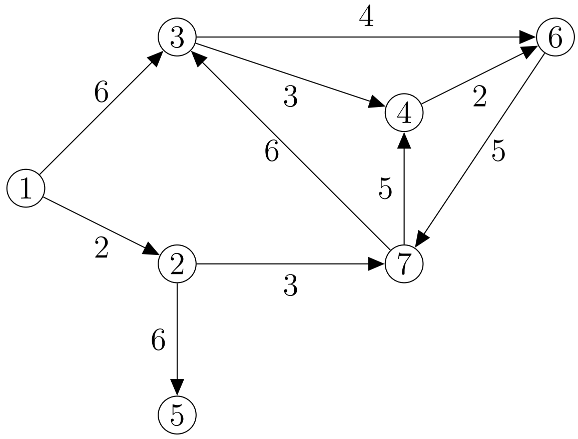
```
Kezd(G,s)
  for(v in V){
    d(v):=INF
    Apa(v):=0
    Kesz(v):=0
  }
  d(s):=0

Kozelit(G,u,v,Q)
  if(d(v)>d(u)+c(u,v)){
    d(v):=d(u)+c(u,v)
    Modosit(Q,v)
    Apa(v):=u
  }
```

```
Dijkstra(G,s)
  Kezd(G,s)
  Letesit(Q: ModPrisor)
  for(v in V)
    SorBa(Q,v)
  while(Elemszam(Q)>0){
    SorBol(Q,u)
    Kesz(u):=1
    for(v in KiEl(G,u)){
      if(Kesz(v)=0)
        Kozelit(u,v)
    }
  }
```

1. Feladat A következő gráfon az 1 csúcsból kiindulva hajtsuk végre a Dijsktra algoritmust:



Megoldás



Csúcs	1	2	3	4	5	6	7
Apa	0	0	0	0	0	0	0
d	0	∞	∞	∞	∞	∞	∞
Kész	✗	✗	✗	✗	✗	✗	✗

Q	1	2	3	4	5	6	7

Minimális feszítőfák

Feszítőfa: minden csúcsot érintő, összefüggő, körmentes élhalmaz

Legyen $G = (V, E, c)$, $c : E \to \mathcal{R}^+$ egy súlyozott irányítatlan gráf. Terjesszük ki a súlyfüggvényt a $T \subseteq E$ élhalmazokra: $C(T) = \sum_{(u,v)\in T} c(u, v)$. (Tehát egy élhalmaz súlya a benne lévő élek összsúlya.)

Az $F = (V, T)$ gráf minimális feszítőfája G-nek, ha

- F feszítőfája G-nek, és
- $C(T)$ minimális

Kruskal algoritmus

Algoritmus:

```
Kruskal(G,w)
  Letesit(A: halmaz)
  for(v in V)
    Halmazt-Keszit(v)    //Kezdetben minden pont egy fa
  rendezzuk E eleit w szerint novekvo sorrendbe
  for((u,v) in E) a suly szerinti sorrendben
    if(Halmazt-Keres(u)!=Halmazt-Keres(v))
      A:=A ∪ {(u,v)}
    Egyesit(u,v)
```

- Minimális fák erdejét tároljuk
- Minden lépésben a legkisebb, két fát összekötő élt húzzuk be (egyesítjük egyetlen fává a két fát)
- Mohó algoritmus
- Megvalósítása: Union-Find adattípussal.
- Futásidő: $O(|E|\log|E|)$

Prim algoritmus

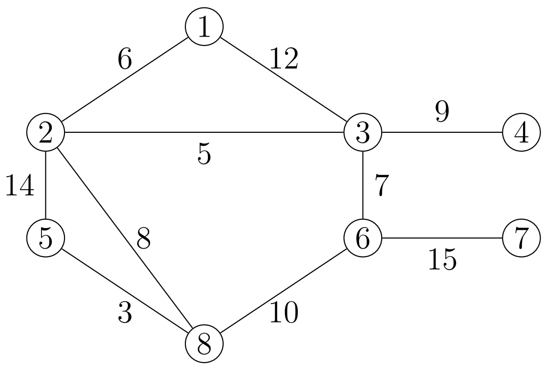
Algoritmus:

```
Prim(G,c,r)
  for(v in V){
    d(v):=INF
    Apa(v):=0}
  d(r):=0
  Letesit(Q: ModPrisor)
  for(v in V){ SorBa(Q,v) }
  while(Elemszam(Q)>0){
    SorBol(Q,u)
    for(v in KiEl(G,u)){
      if(c(u,v)<d(v)){
        Apa(v):=u
        d(v):=c(u,v)
```

```
            Modosit(v)}}}
```

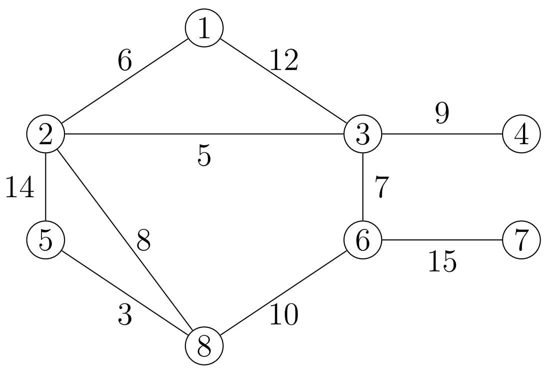
- Egyetlen fát növesztünk
- Tetszőleges gyökérpontból indulva
- Minden lépésben új csúcsot kötünk be a fába
- Legolcsóbb éllel elérhető csúcsot választjuk
- Mohó algoritmus
- Megvalósítása: Bináris kupac adattípussal.
- Futásidő: $O(|E| \log |V|)$

2. Feladat Hajtsuk végre a következő gráfra a Kruskal és Prim algoritmusokat! A Prim-et indítsuk a 6 csúcsból!



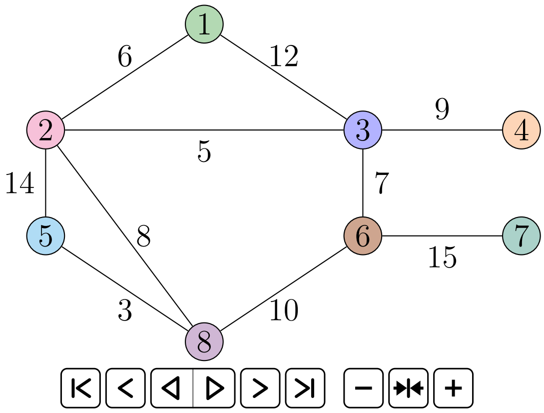
Megoldás

A Prim algoritmus megoldása:



Csúcs	1	2	3	4	5	6	7	8
Apa	0	0	0	0	0	0	0	0
d	∞	∞	∞	∞	∞	∞	∞	∞

A Kruskal algoritmus végrehajtása:



Érdekesség: ha tudni akarod, mi az a Union-Find adattípus és mikor szokás használni, kattints a nyuszira: