

7.hét

Gazdaságos favázak keresése: Prim és Kruskal algoritmusok

Prim algoritmus:

A Prim algoritmus egy olyan mohó algoritmus amellyel egy összefüggő súlyozott gráf minimális feszítőfáját lehet meghatározni. Egy csúcsból indulva növekszik amíg az összes csúcsot el nem éri. Lépései:

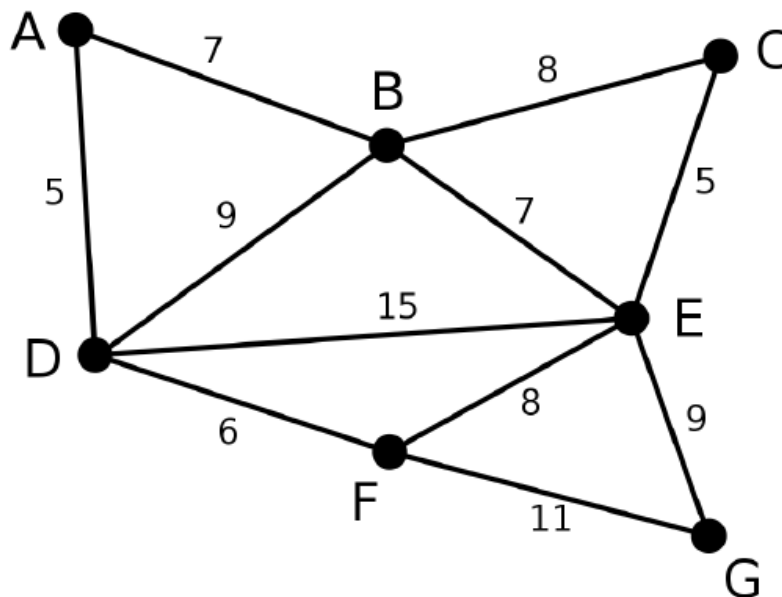
1. Válassz egy tetszőleges csúcsot, ez lesz a minimális feszítőfa első eleme
2. Mindaddig, amíg a kiindulási gráfunknak van olyan csúcsa amely még nem szerepel a minimális feszítőfában, addig végezd el az alábbi lépéseket:

-Válasszuk ki a minimális feszítőfa csúcsa(i) és a fában **még nem szereplő** csúcsok között futó azon élt, amelynek a legkisebb a költsége.

Másképpen: Egy tetszőleges csúcsból indulva elkezdjük felépíteni a fánkat, mindig a legkisebb költségű éleket behúzva úgy hogy a behúzott él **ne alkosson kört**.

FONTOS: amennyiben minden lépésben csupán egy választási lehetőségünk van, tehát egy legkisebb súlyú él, akkor biztosan csak egy megoldás van. Ha nem, akkor lehet akár több is. (ha minden út költsége különbözik, akkor a megoldás egyértelmű)

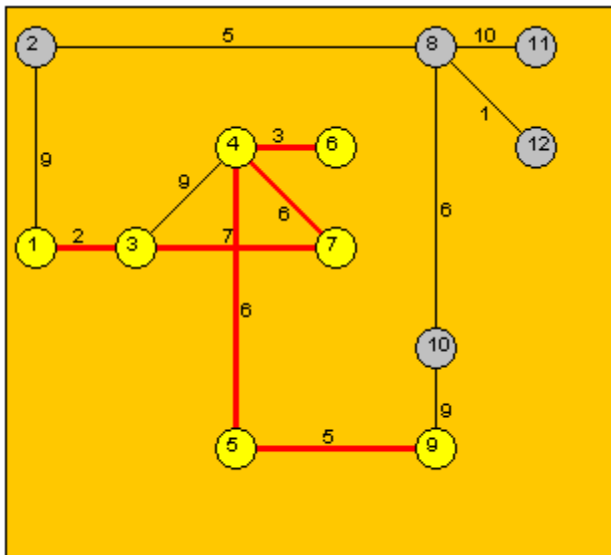
1., Keresd meg a minimális feszítőfát Prim algoritmussal!



1. Válasszuk először a D csúcsot, ez lesz a fánk első eleme
2. Adjuk hozzá a legkisebb súlyú élet AD-t

3. Adjuk hozzá a legkisebb súlyú élet DF-et
4. Adjuk hozzá a legkisebb súlyú élet AB-t
5. Adjuk hozzá a legkisebb súlyú élet BE-t
6. Adjuk hozzá a legkisebb súlyú élet EC-t
7. Adjuk hozzá a legkisebb súlyú élet EG-t

2., Prim algoritmus: Ez egy jó példa arra, ha esetleg két különböző megoldást is kaphatunk, de az útiköltség ugyan annyi lesz mindkét esetben:



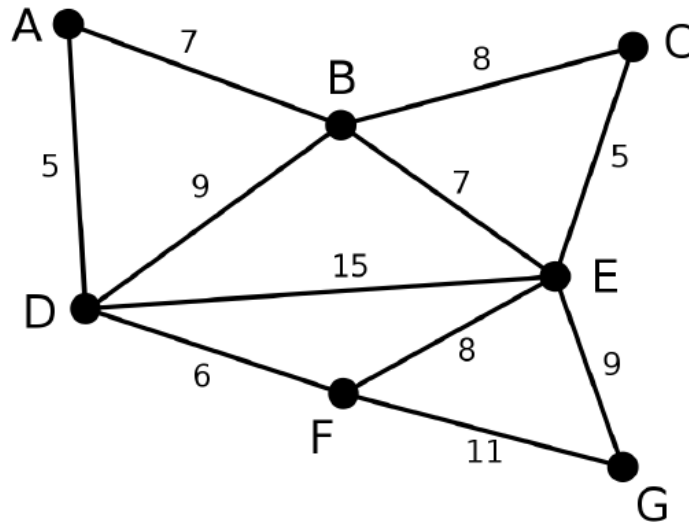
Kruskal algoritmus:

A Kruskal-algoritmus egy súlyozott gráfokat feldolgozó mohó algoritmus. Ha a gráf összefüggő, akkor minimális feszítőfa megalkotására szolgál, ha nem, akkor minimális feszítőerdőt hoz létre. Lépései:

0. Állítsuk a költségük szerinti növekvő sorrendbe az éleket
1. Válasszuk ki a legkisebb súlyú élt
2. Amennyiben az él nem alkot kört a készülő minimális feszítőfánk gráfjában/részgráfjában akkor húzzuk be az élt. (nem kell hogy az él illeszkedjen az készülő fánkra, lehet tőle külön álló is)
3. Ezt folytassuk, amíg az összes csúcs nem szerepel a feszítőfában vagy megvan az $n-1$ db él (n db csúcs van).

FONTOS: amennyiben minden lépésben csupán egy választási lehetőségünk van, tehát egy legkisebb súlyú él, akkor biztosan csak egy megoldás van. Ha nem, akkor lehet akár több is. (ha minden út költsége különbözik, akkor a megoldás egyértelmű)

1., Keresd meg a minimális feszítőfát az alább összefüggő gráfban Kruskal algoritmusának felhasználásával!



Megoldás:

Először rendezzük súlyozás szerint növekvő sorba az éleket.

AD,CE,DF,BE,AB,CB,EF,EG,BD,FG,DE

1. Válasszuk ki a legkisebb súlyú élt, ami jelen esetben vagy az AD vagy a CE, tetszőlegesen választunk kettőjük közül. Legyen az AD az első
2. Válasszuk ki a következő legkisebb súlyú élt CE
3. Válasszuk ki a következő legkisebb súlyú élt DF
4. Válasszuk ki a következő legkisebb súlyú élt BE
5. Válasszuk ki a következő legkisebb súlyú élt AB
6. Válasszuk ki a következő legkisebb súlyú élt CB, de ez kört alkotna, ezért kihagyjuk
7. Válasszuk ki a következő legkisebb súlyú élt EF, de ez kört alkotna, ezért kihagyjuk
8. Válasszuk ki a következő legkisebb súlyú élt EG meg is van az n-1 élünk (n db csúcs van) ezzel készen is vagyunk