**Óravázlat**

**Gráfelméleti alapfogalmak**

Gráf (hálózat) alapfogalmai: gráf, csúcspont (csomópont), él, irányítás nélküli él, irányított él, kezdőpont, végpont, súly, forrás, nyelő, lánc, út, összefüggő gráf, fa, szomszédsági mátrix

Gráf: pontoknak (csomópontok) és pontokat összekötő vonalaknak (élek, ágak) halmaza

Irányítás nélküli él: mindkét irányban lehetséges a közlekedés rajta

Irányított él: csak a nyílheggyel jelölt irányban lehetséges a közlekedés rajta; így van kezdőpontja (forrás) és van végpontja (nyelő)

Súly: számérték, amely jellemzi az adott élt (távolság, idő, költség, kapacitás stb.)

Lánc: olyan élsorozat, amely összeköt két csomópontot

Út: olyan lánc, amelyben egy él legfeljebb egyszer szerepel

Kör: olyan út, amelynek a kezdeti és a végső csomópontja azonos

Összefüggő gráf: ha bármely két csomópontja között létezik legalább egy út

Fa: körmentes összefüggő gráf

Szomszédsági mátrix: sorai egy gráf éleinek kezdőpontjait, oszlopai az élek végpontjait, a mátrix elemei az aktuális él súlyát tartalmazzák

Legrövidebb út: két adott csomópont között létező utak közül a legkisebb összsúlyú

Leghosszabb út: két adott csomópont között létező utak közül a legnagyobb összsúlyú

Forrás: olyan csomópont, amely csak kimenő élekkel rendelkezik

Nyelő: olyan csomópont, amely csak bemenő élekkel rendelkezik

**Gyakorlati alkalmazások irányítás nélküli gráfokban**

**Legrövidebb út keresése két csomópont között Dijkstra címkéző algoritmusával**

Kiindulva az egyik végpontból, lépésenként minden csúcspontot ellátunk először egy ideiglenes, majd egy végleges címkével, amelyek az aktuális csúcspontnak a kezdő csúcsponttól vett távolságát, illetve a legrövidebb távolságát fogja jelenteni, minden lehetséges útvonalon keresztül.

Először a kezdőpont kap egy 0 értékű ideiglenes címkét, amelyet rögtön bekeretezünk, így jelölve végleges voltát. Tehát a kezdőpont önmagától 0 távolságra van.

Utána a kezdőpont szomszédai kapnak ideiglenes címkét, mindegyik olyan értékkel, amely a közte és a kezdőpont közötti él súlyát(távolságát) jelöli. Az ideiglenes címkék közül a legkisebbet (ha több van akkor az egyik legkisebbet) véglegesítjük.

Majd az újonnan véglegesített címkéjű csomópont szomszédaihoz (amelyek még nem rendelkeznek végleges címkével) számítunk ideiglenes címkéket úgy, hogy a jelenlegi végleges címke értékéhez hozzáadjuk az aktuális él súlyát. Ha még nem volt egy szomszéd csomópontnak ideiglenes címkéje, akkor ez lesz az, ha már volt, de kisebb, mint a jelenlegi, akkor marad az előző, ha már volt, de nagyobb, akkor azt lecseréljük (áthúzzuk) a mostani kisebbre.

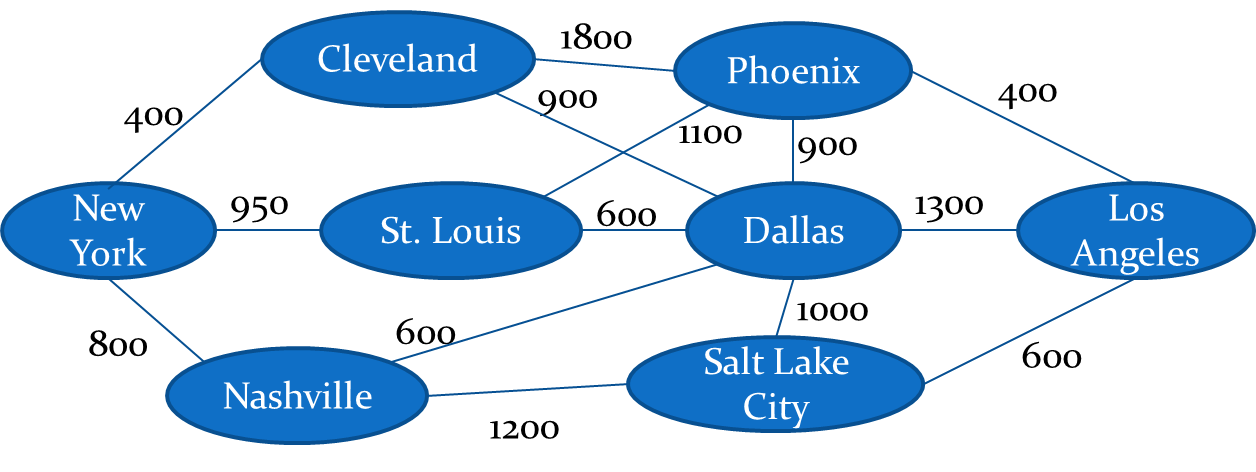
Addig folytatjuk az eljárást, amíg a végpont címkéje véglegessé válik. Ez a számérték mutatja a legrövidebb út hosszát.

A legrövidebb utat (utakat) a végpontból kiindulva kell visszakeresni a végleges címkék keletkezési irányai mentén.

**Példa**

Teherautóval kell utazni New Yorkból Los Angelesbe. Különféle útvonalak léteznek. Az egyes élekhez rendelt szám az az üzemanyag-mennyiség (gallonban), amelyre a teherautónak szüksége van az út megtételéhez.

Alkalmazza Dijkstra algoritmusát annak a meghatározására, hogy melyik útvonalat kell választani a miimális üzemanyag-mennyiség felhasználásához. Hány gallon ez a mennyiség?



**Megoldás**

1. Lépés: Kihelyezzük NY ideiglenes és egyben végleges címkéjét

1800

1100

400

400

900

900

0

1300

950

600

1000

600

600

800

1200

1. Lépés: Ideiglenes címkék CL, St. L, NV városokhoz, véglegeset CL városhoz

400

1800

1100

400

400

900

900

950

0

1300

950

600

1000

800

600

600

800

1200

1. Lépés: Ideiglenes címkék PH, DA városokhoz, véglegeset NV-nek

2200

400

1800

1100

400

400

900

900

1300

950

0

1300

950

600

1000

800

600

600

800

1200

1. Lépés: Ideiglenes címke SLC városnak, végleges St. L-nak

2200

400

1800

1100

400

400

900

900

950

1300

0

1300

950

600

1000

2000

800

600

600

800

1200

1. Lépés: PH ideiglenes címkéjét lecseréljük egy kisebbre, végleges címke DA-nak

2050

2200

400

1800

1100

400

400

900

900

1300

950

0

1300

950

600

1000

2000

800

600

600

800

1200

1. Lépés: Ideiglenes címke LA-nek, végleges SLC-nek

2050

2200

400

1800

1100

400

400

900

900

2600

1300

950

0

1300

950

600

1000

2000

800

600

600

800

1200

1. Lépés: Végleges címke PH városhoz, LA ideiglenes címkéjét lecseréljük egy kisebbre

2050

2200

400

1800

1100

400

2450

400

900

900

2600

1300

950

0

1300

950

600

1000

2000

800

600

600

800

1200

1. Lépés: LA végleges címkét kap

2050

2200

400

1800

1100

400

2450

400

900

900

2600

1300

950

0

1300

950

600

1000

2000

800

600

600

800

1200

Megoldás: a minimális üzemanyagfogyasztás (2450 gallon) a következő útvonalon lesz lehetséges

New York (NY) → St. Louis (St. L) → Phoenix (PH) → Los Angeles (LA)

**Minimális kifeszítő fa keresése Prim-, illetve Kruskal algoritmusokkal**

A minimális kifeszítő fa problémája olyan élek kiválasztását jelenti, amelyek az összes csomópontot összekötő fastruktúrát alkotnak úgy, hogy azok súlyainak összege minimális. Az általunk használt két mohó algoritmus lényege, hogy egyenként kiválasztjuk a következő módon:

Kruskal algoritmussal kiválasztjuk mindig a lehetséges legrövidebbet (vagy az egyik legrövidebbet), vigyázva, hogy az újonnan kiválasztott él ne alkosson kört az addig kiválasztottakkal,

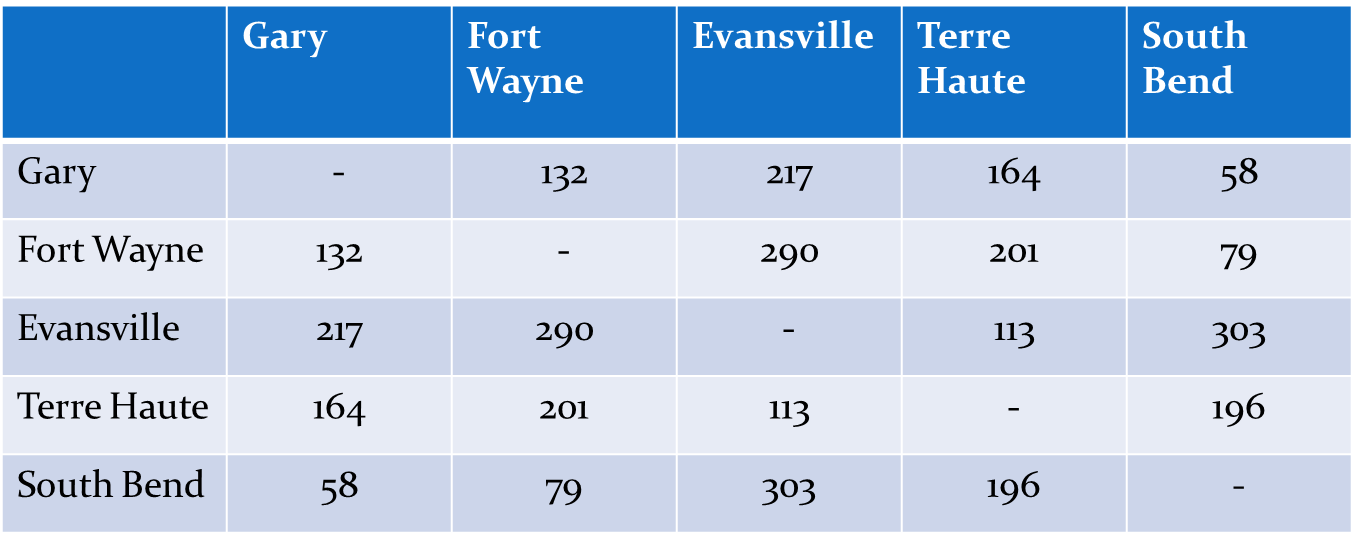
Prim algoritmussal kiindulunk egy megadott vagy tetszőlegesen kiválasztott csomópontot és a legközelebbi egyik szomszédjával összekötő élt, majd a kiválasztott él két csúcspontja szomszédai közül a legközelebbivel összekötő élt, és mindig aa már kiválasztott élek csúcspontjainak összes szomszédai közül az egyik legrövidebbet, vigyázva, hogy a kiválasztott élek ne alkossanak kört,

és midezt addig folytatjuk, míg a kiválasztott élek összekötik az összes csomópontot (n csomópont esetén n – 1 élre van szükség a kifeszítő fához).

**Példa**

A távolságot (mérföldben) Gary, Fort Wayne, Evansville, Terra Haute és South Bend indianai városai között a következő táblázat mutatja. Építeni kell egy, az összes várost összekötő állami útrendszert. Tegyük fel, hogy nem lehet utat építeni Gary és Fort Wayne között, és nem lehet olyan utat sem építeni, amely összeköti South Bendet Evansville-el.

Mekkora a minimális úthossz?



Rajzoljuk meg a probléma gráfját, majd a Kruskal, illetve a Prim módszerek segítségével keressünk minimális kifeszítő fát.

**Megoldás**

Megrajzoljuk a feladat gráfját.

58

164

79

196

201

113

290

217

A piros vonalak a tiltott útvonalakat jelölik.

A minimális kifeszítő fa élei, illetve az össztávolság:

Prim algoritmusát használva (amennyiben Gary-ből indulunk) az alábbi sorrendet adja

G-SB, SB-FW, G-TH, TH-E , a megfelelő távolságokkal

58 + 79 + 164 + 113 = 414 (mérföld)

Kruskal algoritmusát használva az alábbi sorrendet adja

G-SB, SB-FW, TH-E, G-TH , a megfelelő távolságokkal

58 + 79 + 113 + 164 = 414 (mérföld)

**Gyakorlati alkalmazások irányított gráfokban**

**Leghosszabb út irányított gráfban (szomszádségi mátrixos módszer)**

Az algoritmus első lépéseként elkészítjük az irányított gráf szomszédsági mátrixát. Ilyenkor a gráf pontjainak a mátrix sorai, illetve oszlopai felelnek meg.

Felső háromszög mátrixot kapunk a szomszédsági mátrixban, ha a mátrix elkészítése során a pontokat úgynevezett előre mutató sorrendben tekintjük. Az előre mutató sorrend olyan sorrend, ahol a gráf minden éle előrefelé mutat, azaz a sorrendben a gráf minden élének kezdőpontja megelőzi végpontját.

A leghosszabb út oszlopába beírjuk először az első helyre a 0 értéket, majd fentről lefelé haladva lépésenként kiszámítjuk, hogy minden egyes csomópontnak mekkora a legnagyobb távolsága a kezdőponttól, illetve melyik megelőző csomópontból valósult ez meg. Az aktuális csomópont oszlopába beírt értékeket egyenként hozzáadjuk a leghosszabb út oszlopába előzőleg kiszámolt megfelelő távolsághoz és a legnagyobb értéket tartjuk meg, és mellé írjuk annak a csomópontnak a betűjelét, ahonnan ez a maximális érték megvalósult.

**Példa**

Keressük meg a C és G csomópontokat összekötő leghosszabb utat, illetve annak hosszát.

8

2

**A**

**B**

5

5

6

3

**C**

8

**E**

5

8

**F**

2

**D**

5

**G**

5

**Megoldás**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kezd./Vég. | C | A | B | E | D | F | G | Leghosszabb út |
| C |  | 2 |  | 6 | 8 |  |  | 0 |
| A |  |  | 8 | 5 |  |  |  | 2C |
| B |  |  |  | 3 |  | 5 |  | 10A |
| E |  |  |  |  | 2 | 8 | 5 | 13B |
| D |  |  |  |  |  |  | 5 | 15E |
| F |  |  |  |  |  |  | 5 | 21E |
| G |  |  |  |  |  |  |  | 26F |

Tehát a leghosszabb út C és G között **C→A→B→E→F→G**, és hossza **26**.

**Maximális folyam problémája irányított gráfban (hálózatban)**

Megoldási módszerünket csak olyan irányított gráfokra alkalmazzuk, amelyeknek egyetlen forrásuk és egyetlen nyelőjük van. Az élekben a megadott irányban áramlás folyik a súlynak (kapacitásnak) megfelelő mértékben. Amennyi a kedőponton befolyik, annyi folyik ki a végpontban (tehát megmaradó áramlásról beszélünk).

Cél az, hogy kiszámoljuk a hálózaton keresztülfolyó áramlás kapacitásának maximumát egységnyi időtartam alatt.

**Példa**

Adott egy irányított gráf, amelyben egy vezetérendszer csőhálózatát látjuk lerajzolva. A csomópontok elágazások, az élek csövek, amelyeknek súlya megadja az adott cső kapacitását, azaz mennyi folyadék tud időegység alatt átjutni rajta a megadott irányban (m3/mp). A vezeték-hálózatnak van egy kezdőpontja (forrás vagy csap) és egy végpontja (nyelő vagy kifolyó). Minden vezeték csak a nyíl által megadott irányban használható.

Példánkban van 5 csomópont (forrás, 3 elágazás, nyelő) és 7 cső a megadott irányokkal és súlyokkal.

A

C

B

1

4

3

5

6

E

D

2

6

A maximális folyam problémáját megoldhatjuk elemi módon, szabad kapacitású útvonalak sorozatos kiválasztásával vagy minimális vágás módszerével, de lineáris programozási modell felírásával is lehetséges. Ezúttal a lineáris programozási módszert fogjuk alkalmazni.

Ehhez be kell vezetni egy mesterséges élet, amely a nyelőből visszavezet a forrásba, amely maximális kapacitás értékének meghatározása lesz a cél az LP feladatban.

A

C

B

1

4

3

5

6

E

D

2

6

X0

Vezessünk be döntési változókat az alábbiak szerint.

xij = a hálózat (i,j) élén keresztül haladó folyadék mennyisége időegység alatt

x0 = a nyelőbe érkező folyadékmennyiség

A korlátozó feltételekben figyelembe kell venni, hogy

minden élen a kapacitásnak megfelelő mennyiség haladhat át legfeljebb

minden csúcsban a bemenő folyam meg kell egyezzen a kimenő folyammal

A matematikai modell a következő lesz.

|  |  |
| --- | --- |
| Nemnegatívitás | xAB, xAD, xBC, xBE, xCD, xCE, xDE, x0∈ N |
| AB él | xAB ≤ 6 |
| AD él | xAD ≤ 4 |
| BC él | xBC ≤ 2 |
| BE él | xBE ≤ 3 |
| CD él | xCD ≤ 5 |
| CE él | xCE ≤ 1 |
| DE él | xDE ≤ 6 |
| A csúcs | x0 = xAB+xAD |
| B csúcs | xAB = xBC+xBE |
| C csúcs | xBC = xCD+xCE |
| D csúcs | xAD+xCD= xDE |
| E csúcs | xBE+xCE+xDE=x0 |
|  |  |
| Célfüggvény | x0 → MAX |

Solveres megoldás

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | xAB | xAD | xBC | xBE | xCD | xCE | xDE | x0 | SzorzatÖsszeg | RelációJel | JobbOldal |
| AB él | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | "≤" | 6 |
| AD él | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | "≤" | 4 |
| BC él | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | "≤" | 2 |
| BE él | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 | "≤" | 3 |
| CD él | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | "≤" | 5 |
| CE él | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | "≤" | 1 |
| DE él | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | "≤" | 6 |
| A csúcs | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | "=" | 0 |
| B csúcs | 1 | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | "=" | 0 |
| C csúcs | 0 | 0 | 1 | 0 | -1 | -1 | 0 | 0 | 0 | "=" | 0 |
| D csúcs | 0 | 1 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | "=" | 0 |
| E csúcs | 0 | 0 | 0 | 1 | 0 | 1 | 1 | -1 | 0 | "=" | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |
| CF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 9 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| MO | 5 | 4 | 2 | 3 | 2 | 0 | 6 | 9 |  |  |  |

A megoldás tehát 9 m3/mp lesz.

Útvonalak kapacitásokkal, amelyeken a maximális mennyiség átjut a hálózaton:

A – B – E = 3

A – B – C – D – E = 2

A – D – E = 4

**Hálótervezés kritikus út kereséssel**

A tevékenységi háló egy olyan irányított gráf, ahol az élek valamilyen tevékenységet, munkát szimbolizálnak. A háló logikai rendbe állítja az elvégzendő feladatokat. A tevékenységeken csak a nyilak szerinti sorrendben lehet végighaladni. Például házépítés során az alap elkészítése csak az ásás után történhet, a falazás csak az alapozás után és így tovább. A hálóban a nyilak vége jelzi a tevékenységek végét, a másik vége a tevékenységek elejét.

A kritikus út tevékenységeit kritikus tevékenységeknek nevezzük, mivel ezekkel várakozni nem lehet. Azokkal a tevékenységekkel viszont, amelyek nem elemei egyetlen kritikus útnak sem, várakozhatunk.

A kritikus útban szereplő csúcsok legkorábbi és legkésőbbi értéke megegyezik. A legkésőbbi és legkorábbi értékekből lehet következtetni a tevékenységek várakozási idejére.

**Példa**

Adott egy összetett feladat, amely egymásra épülő tevékenységekre bontható. A projekt kezdetét jelöljük A-val, a végét pedig E-vel. A tevékenységeket egy irányított gráf éleiként tudjuk ábrázolni, amelyeknek súlyai az elvégzésükhöz szükséges időtartamokat jelentik napokban megadva. Az egymásra épülő tevékenységeket csak a megadott sorrendben lehet elvégezni, de a „párhuzamosan futókat” lehet akár egyidőben is, több személy vagy csapat által megoldani.

A csomópontok eseményeket jelentenek: kezdete azoknak a tevékenységeknek, amelyek belőle indulnak, illetve vége azoknak a tevékenységeknek, amelyek bele érkeznek. Egy csomópontból csak akkor lehet továbbindulni, ha az oda befutó tevékenységek már mind befejeződtek.

A

C

B

9

3

8

5

5

E

D

6

3

**Megoldás**

Először bevezetünk döntési változókat:

xi: az i-edik csomópont (esemény) bekövetkezésének ideje (napokban mérve a kezdéstől számítva)

A matematikai modell:

|  |  |
| --- | --- |
| Nemnegatívitás | xA, xB, xC, xD, xE∈ N |
| AB él | xB-xA ≥ 5 |
| AD él | xD-xA ≥ 3 |
| BC él | xC-xB ≥ 6 |
| BE él | xE-xB ≥ 8 |
| CD él | xD-xC ≥ 5 |
| CE él | xE-xC ≥ 9 |
| DE él | xE-xD ≥ 3 |
|  |  |
| Célfüggvény | xE-xA → MIN |

Solveres megoldás:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | xA | xB | xC | xD | xE | SzorzatÖsszeg | RelációJel | JobbOldal |
| AB él | -1 | 1 | 0 | 0 | 0 | 5 | "≥" | 5 |
| AD él | -1 | 0 | 0 | 1 | 0 | 17 | "≥" | 3 |
| BC él | 0 | -1 | 1 | 0 | 0 | 6 | "≥" | 6 |
| BE él | 0 | -1 | 0 | 0 | 1 | 15 | "≥" | 8 |
| CD él | 0 | 0 | -1 | 1 | 0 | 6 | "≥" | 5 |
| CE él | 0 | 0 | -1 | 0 | 1 | 9 | "≥" | 9 |
| DE él | 0 | 0 | 0 | -1 | 1 | 3 | "≥" | 3 |
|  |  |  |  |  |  |  |  |  |
| CF | -1 | 0 | 0 | 0 | 1 | 20 |  |  |
|  |  |  |  |  |  |  |  |  |
| MO | 0 | 5 | 11 | 17 | 20 |  |  |  |

Tehát az egész projekt megoldásához legalább 20 napra van szükség. Ennek hátterében a forrást (A) a nyelővel (E) összekötő utak leghosszabbja áll, amit kritikus útnak nevezünk. Mivel a kritikus úton lévő tevékenységeket csak a megadott sorrendben és csak egymás után lehet elvégezni, ezért nincs lehetőség 20 napnál rövidebb idő alatt befejezni a munkálatokat.

A kritikus út: A → B → C → E .

A

C

B

9

3

8

5

5

E

D

6

3

