Bevezetés a számelméletbe 2.

# Gráfelméleti alapfogalmak

## Definíciók

* Gráf
  + Egy rendezett (V, E) halmaz, ahol V nem üres halmaz, E pedig V-ből alkotott rendezett vagy rendezetlen párokat tartalmaz
* Egyszerű gráf
  + Nem tartalmaz hurokélet (olyan él melynek 2 végpontja ugyanaz a csúcs)
  + Nem tartalmaz többszörös élet (2 olyan nem hurokél, melynek végpontjai megegyeznek)
* Fokszám
  + d(v)
* Maximális fokszám
  + (G)
* Teljes gráf
* Összefüggő gráf
* Izomorf gráfok
  + G és G’ izomorfak, ha létezik olyan bijekció, hogy G’ pontjai pontosan akkor vannak összekötve, ha G-beli párjaik is
* Komplementer gráf
* Élsorozat
* Séta
  + Olyan élsorozat, melyben minden él különböző
* Körséta
  + Olyan séta, amiben a kezdeti és a végpont megegyezik
* Út
  + Olyan séta amiben nincs csúcsismétlődés
* Kör
  + Olyan körséta amiben nincs csúcsismétlődés
* Fa
  + Körmentes egyszerű gráf
* Részgráf
* Feszített részgráf
  + A kiválasztott pontok közé az összes eredeti élet behúzzuk
* Feszített részfa
* Euler-séta
  + Olyan séta, mely minden élet tartalmaz
* Euler-körséta
* Hamilton-kör
  + Olyan kör ami minden csúcsot tartalmaz

## Tételek

* |V| + 1 = |E|
* Egy fába behúzunk még egy élet, azzal pontosan egy kör lesz benne
* Ekvivalencia Euler körséta létezésére (összefüggő, egyszerű gráfban)
  + Minden fokszám páros
  + BIZONYÍTÁS
    - Ha van benne Euler körséta, akkor minden fokszám páros (minden csúcsba annyiszor mentem be, ahányszor ki)
    - Tegyük fel, hogy minden G-nél kisebb élszámú páros fokszámú gráfról már tudjuk, hogy van benne Euler körséta
    - Ha minden fokszám páros, akkor biztosan találok egy kört amin végig tudok menni: úgy megyek végig a csúcsokon, hogy minden élet csak max egyszer érintek
      * Ha ezt már nem tudom tovább csinálni, mert elakadtam, akkor a végpontban vagyok, különben tovább tudtam vna menni, hiszen minden fokszám páros
    - Hagyjuk el ezt a kört, a gráfban most biztosan páros fokszámú kisebb élszámú komponensek vannak, amikről „tudjuk”, hogy van bennük Euler-körséta
    - Emiatt G-ben is van, hiszen ha elindulok a megtalált körömön és átmennék egy ilyen komponensen, akkor végig megyek a komponens Euler körsétáján, és úgy megyek tovább a körömön
* Ekvivalencia Euler séta létezésére (egyszerű gráfban)
  + 0 vagy 2 db ptlan fokú csúcsa van
  + BIZONYÍTÁS
    - Ha van Euler sétám akkor a 2 vége közé behúzok egy élet, amivel van Euler körsétám
    - Erre a gráfra igaz az előbbi tétel, ennek az élnek az elhagyásával pedig triviálisan következik ez a tétel
* Szükséges feltétel Hamilton-kör létezésére (egyszerű gráfban)
  + Nem lehet úgy k csúcsot elhagyni, hogy k-nál több komponensre essen
  + BIZONYÍTÁS
    - Rajzoljunk be egy Hamilton kört, hagyjunk el k pontot és lássuk be, hogy legfeljebb k komponensre esik szét
    - Ha fel van rajzolva a Hamilton kör (amiben az összes csúcs ugyebár benne van), elhagyhatok k darabot, mellyel az elhagyott csúcsok közötti ívek összefüggőek maradtak
    - Ezzel szét tudjuk szedni a gráfot k komponensre, de ennél csak kevesebbre lehet, ha szomszédos csúcsokat hagyunk el a Hamilton körből, vagy ha vannak még élek amik „összetartják” a komponenseket
* Elégséges feltételek Hamilton-kör létezésére (egyszerű gráfban)
  + Minden csúcs fokszáma n/2 vagy nagyobb (Dirac)
    - BIZONYÍTÁS
      * A következő tételből egyértelműen következik
  + Nemszomszédos csúcsok fokszámösszege >= n (Ore)
    - BIZONYÍTÁS
      * Tegyük fel indirekt, hogy adott egy gráf melyben d(x) + d(y) >= n (x és y nem szomszédosak) de nincs ebben a gráfban Hamilton-kör
      * Adjunk hozzá éleket addig, amíg már nem lehet olyan élet belehúzni, hogy az ne alkosson Hamilton-kört
      * Ebben a gráfban választok egy x és egy y nemszomszédos csúcsot, melyről evidens, hogy ha behúznám az {x, y} élet, akkor lenne Hamilton kör a gráfban, vagyis ebben a gráfban van egy Hamilton-út
      * Ennek a Hamilton-útnak a 2 végpontja x és y
      * y nem lehet összekötve x egyik szomszédjának a szomszédjával sem (akkor is zárna egy Hamilton-kört) vagyis nem lehet összekötve több mint n – d(x) – 1 ponttal (d(y) < n – d(x) – 1)
      * Ez ellentmond annak, hogy a kezdeti gráfban d(x) + d(y) >= n

# Síkbanrajzolhatóság, színezés

## Definíciók

* Síkbanrajzolhatóság
  + Létezik olyan diagramja, hogy az élei ne metszék egymást
* Tartományok
  + Síkban rajzolt gráfok élei által közrefogott síkrészek
* Gömbre rajzolható
* Pontszínezés
  + Függvény, ami minden ponthoz egy olyan értéket rendel, amilyet a szomszédos pontokhoz nem
* Élszínezés
  + Függvény ami minden **hurokmentes** gráf minden éléhez egy olyan értéket rendel ami különbözik a vele szomszédos élekétől (azoktól az élekétől melyekkel közös végpontja van)
* Kromatikus szám

  + A minimális színt használó pontszínezés
* Élkromatikus szám
* Klikkszám
  + A legnagyobb teljes részgráf csúcsainak száma
  + w(G)
* Mohó színezés
  + Választok előre egy csúcs sorrendet
  + Minden csúcsot a legkisebb lehetséges színnel színezek
* Intervallumgráf
  + Olyan gráf, melyre igaz, hogy csúcsai megfeleltethetőek intervallumoknak, élei pedig az intervallumok közti fedéseknek
* Topologikus izomorfia
  + Ezt nem tudom jól elmagyarázni
* Síkban rajzolt gráf dualásia
  + Az a gráf melynek képzési szabálya a következő: a síkban rajzolt gráf tartományai a duális csúcsai, és azok a csúcsok vannak összekötve amely tartományok szomszédosak

## Tételek

* Ha síkra rajzolható, akkor gömbre is
  + BIZONYÍTÁS
    - Szetereografikus projekcióval minden síkban és gömbre rajzolt gráfnak meg tudunk feleltetni egy másikat
    - Síkról gömbre: Lehelyezünk egy gömböt a síkra, és a síkban rajzolt gráf minden pontját összekötöm az északi pólusával, és a gömbfelülettel való metszéspontját veszem
    - Gömbről síkra ugyanígy
* Minden síkban rajzolt **összefüggő** gráfra c + t = e + 2 (Euler-tétel)
  + t a tartományok száma
  + BIZONYÍTÁS
    - Adott c csúcs, t tartomány és e él
    - Minden kör a síkot 2 részre osztja
    - Vegyük a gráf egyik körét, hagyjuk el az egyik élét, ezzel t és e 1-el csökken
      * c + t – e épp ezért nem változik
    - Ezt folytatom addig amíg egy feszítőfát nem kaptam
    - Ekkor t = 1 és e = c - 1
* Ha egy síkban rajzolt **összefüggő, egyszerű** gráfra igaz, hogy nem húzható belé olyan él, hogy síkbanrajzolható maradjon, akkor
  + t = 2 / 3 \* e
  + e = 3c – 6
  + Ehhez legalább **3 pontja van**
  + BIZONYÍTÁS
    - Ha nem húzható bele több él, akkor minden tartományt pontosan 3 él határol
    - Ezek a 3 élek azonban 2 tartományt határoznak meg, ebből t = 2/3 e
    - Ebből Euler tétellel felírható, hogy e = 3c - 6
* Egyéb esetben
  + e <= 3c – 6
    - BIZONYÍTÁS
      * Előző tételből triviális
  + Ha nincs benne 3szög akkor e <= 2c – 4
    - BIZONYÍTÁS
      * Az előzőhöz hasonlóan
* Ekvivalencia síkbanrajzolhatóságra (Kuratowski-tétel)
  + A gráfnak nincs topologikusan izomorf RÉSZgráfja K3, 3-mal vagy K5-tel
  + BIZONYÍTÁS
    - Ha van benne, akkor nem síkbarajzolható
    - Be kell látni: K3,3 és K5 nem síkbarajzolhatóak
    - K5-re nem teljesül az Euler tétel (e <= 3c - 6)
    - K3,3-ban nincs 3szög (hiszen semmilyen páratlan kör nincs benne) vagyis nem tesz eleget az Euler tételnek (e <= 2c - 4)
* >= w(G)
  + BIZONYÍTÁS
    - Triviálisan igaz, a legnagyobb klikk minden csúcsát különböző színűre kell színezni
* ha G egyszerű, összefüggő, nem teljes és nem páratlan kör (Brooks)
* ha G egyszerű, összefüggő és nem reguláris (gyenge Brooks)
  + BIZONYÍTÁS
    - Mohó színezéssel képesek vagyunk kiszínezni A(G)-vel minden gráfot, csak kell egy jó sorrend
    - A v1, v2, v3… vn akkor lesz jó sorrend, ha bármelyik tagjának A(G)-1-nél kevesebb nála kisebb indexű (már kiszínezett) szomszédja van
      * Vagyis az utolsó tag nem A(G) fokszámú, és minden előtte levőnek van nála nagyobb indexű szomszédja
    - Ilyen sorrendet ad az alábbi algoritmus:
      * Kiválasztunk egy tetszőleges „vn” nem A(G) fokszámú csúcsot (ilyen van, hiszen nem reguláris a gráf)
      * Vesszünk egy feszítőfát G-ben és elkezdjük törölni a vn-től különböző leveleit
      * A törlés sorrendje épp egy nekünk kellő sorrendet határoz meg
* ≥(G) (Vízing)
  + A felső becslés csak egyszerű gráfokra igaz
  + BIZONYÍTÁS (csak alsó határra)
    - triviális
* ha G páros (König)
  + BIZONYÍTÁS 1: minden reguláris páros gráfban van teljes párosítás
    - Ellenőrizzük a Hall feltételt
    - A-ból választunk k pontot, melyekből k \* r él megy ki
    - Mivel B-ben is minden fokszám r, ezért ezen élek végén legalább kdb pontnak kell lennie
    - Vagyis |N(X)| >= |X|
  + BIZONYÍTÁS
    - Minden G gráf egy reguláris páros gráf részgráfja
      * Pl. Ha felvesszük G-t és inverzét: G’t, akkor az ezekből alkotott H gráf (amiben a színosztályok A’ + B és A + B’ ill. minden u és u’ közé húzunk még egy élet) egy reguláris páros gráf, melynek G egy részgráfja
    - Ebben a gráfban tudjuk, hogy van teljes párosítás
    - A teljes párosítás éleit színezzük egy színre majd töröljük, ezzel a gráf még mindig páros és reguláris, vagyis van benne teljes párosítást, úgyh megismételjük a műveletet
    - Ezzel delta(G) színnel kiszíneztük a nagyobb gráfot, tehát a részgráfját is
* Létezik w(G) = 2 gráf bármilyen -hez (Mycielski)
  + Minden eddigi pontot megkettőzünk és összekötjük azokkal akikkel az eredeti pont is össze volt
  + Plusz egy ponthoz hozzákötjük az összes új pontot
  + BIZONYÍTÁS
    - K=2-re igaz
    - Tfh K=n-re igaz
    - K = n+1 esetén
      * A gráf még mindig nem tartalmaz 3szöget
        + Ha tartalmazna 3-szöget az azt jelenti, hogy az új csúcsok 3szöget alakítottak ki, vagyis az eredeti „képük” szomszédos 2 olyan csúccsal amik egymással is szomszédosok
        + Ez lehetetlen, hiszen az indukciós feltevésünk, hogy ilyen nincs
      * A gráf csak n+1-színezhető
        + Legyenek Gn csúcsai v1, v2 stb., ezek képei u1, u2 stb. és az 1 új csúcs w.
        + Indirekt tegyük fel, hogy n-színezhető
        + Legyen w piros, vagyis egyik u csúcs sem piros
        + Most vegyük v-k közül az összes pirosat, és színezzük az u-beli párjuk színére

Ennek a képzési szabály szerint egy helyes színezésnek kell lennie

* + - * + Ezzel viszont átszíneztük az összes piros csúcsot v-kben vagyis Gn-t kiszíneztük n-1 színnel, ami lehetetlen
* Van olyan csúcssorrend, hogy a mohó színezés kromatikus számmal színezzen
  + Intervallumgráf esetén ez a sorrend az intervallumok kezdőpontjainak sorrendje
  + BIZONYÍTÁS
    - Tulajdonképpen már beláttuk a gyenge Brooks tételnél nem reguláris gráfokra
* Intervallumgráfok esetén
  + BIZONYÍTÁS
    - Elkezdjük mohón színezni a kezdőpontjuk szerinti sorrendben
    - Ha szükség lenne w(G) + 1 színre, az csak akkor lenne lehetséges, ha az adott intervallum kezdőpontja fedésben w(G)db másik intervallummal, ami lehetetlen, hiszen az azt jelentené, hogy van w(G)-nél nagyobb klikk

# Páros gráfok és tulajdonságok

## Definíciók

* Lefogó pontok
  + Nincs olyan él, melynek egyik végpontja ne lenne benne ebben a halmazban
* Lefogó élek
  + Nincs olyan pont mely ne lenne az élek egyik végpontja
* Független pontok
  + Olyan ponthalmaz melyben egyik 2 pont között sem fut él
* Független élek
  + Olyan élek melyeknek nincs közös végpontja

|  |  |  |
| --- | --- | --- |
|  | Max független | Min lefogó |
| Pont |  |  |
| Él |  |  |

* Páros gráf
  + Olyan gráf, melynek csúcsai két csoportra oszthatóak, és nincs olyan él mely azonos csoporton belüli pontok közt futna
* Párosítás
  + Független élek halmaza
* Teljes párosítás
  + Egyszerre független és lefogó élek halmaza

## Tételek

* Egy gráf akkor és csak akkor páros, ha nincs benne páratlan kör
  + Ha páros, akkor nincs benne páratlan kör, ez triviális
  + Ha csak páros kör van benne, akkor választunk egy csúcsot, A halmazba tesszük, a szomszédait B-be, az ő szomszédjait ismét A-ba stb. (Ha nem összefüggő, akkor ezt minden komponensre megcsináljuk)
  + Az így kapott elosztás jó
  + Egyenlőség páros gráf esetén (König 1)
  + BIZONYÍTÁS
    - Már csak az alfa db független pont lefogásához szükség van alfa db lefogó élre
  + BIZONYÍTÁS (König)
    - Bebizonyítottuk, hogy v(G) = T(G), Gallai tételekből pedig egyértelmű a bizonyítás
* (Gallai 1)
  + **Hurokmentes gráfokra**
  + BIZONYÍTÁS
    - Ha X a maximális független pontjaim halmaza, akkor G – X a minimális lefogó pontjaim halmaza
    - Ha G – X nem elég a lefogáshoz, akkor van egy élem aminek mindkét végpontja X-ben van, ami lehetetlen
    - Ha G – X-ben volna olyan pont ami már nem is lenne szükséges a lefogáshoz (vagyis G-X nem minimális lefogó halmaz), akkor a fennmaradó 1 pont független lenne X minden pontjától, ami lehetetlen, hiszen X maximális
      * Azért lenne független, mert az X-ben lévő pontokhoz tartozó összes él másik végpontja már le van fogva, vagyis a fennmaradó pont nem lehet összekötve egyikükkel sem
  + Egyenlőség páros gráf esetén (König 2)
    - BIZONYÍTÁS
      * Csak a vdb él lefogásához szükség van vdb pontra
    - BIZONYÍTÁS (König)
      * A páros gráfunkban találtunk egy maximális párosítást, M-et
        + X legyen Ma, U legyen A – X
        + T’ legyen azon B-beli pontok halmaza ahova visz alternáló út U-ból
        + T legyen T’-k M-beli párjai
      * T’ + (X – T) minden élet lefognak (számosságuk M)
        + Általuk nem lefogott él A-ban csak T + U-ból indulhatna
        + U-nak biztosan minden szomszédja T’-ben van, ez T’ definíciójából adódik
        + T-nek is minden minden szomszédja T’-ben van, hiszen T’-ből el lehet jutni T-be, aminek ha lenne azon kívüli szomszédja, akkor oda el lehetne jutni U-ból alternáló úton
      * Mivel M darab ponttal le lehet fogni minden élet, ezért T(G) <= |M|
      * Mivel M-ben minden él független így |M| <= v(G)
      * Ebből pedig T(G) <= és >= v(G) vagyis T(G) = v(G)
* (Gallai 2)
  + Feltéve, hogy **nincs izolált pont**
  + BIZONYÍTÁS
    - vdb él lefog 2v pontot, a maradék n – 2v pont lefogható n – 2v éllel
      * Ebből Ro <= n – 2v + v = n – v
      * **Ro + v <= n**
    - Másrészt ha F egy minimális lefogó élhalmaz, akkor abban nincs se kör, se 3 hosszú út
      * F tehát diszjunkt csillagok uniója
    - Ha F kdb ilyen csillagból áll, akkor egyrészt v >= k, másrészt F elemszáma épp n – k, hiszen minden n csúcshoz saját lefogó él tartozik, kivéve a kdb „csillag közepéhez”
      * Vagyis **Ro + v >= k + n – k = n**
    - Ebből a két egyenlőségből Ro + v = n
* Ekvivalencia páros gráfban teljes párosítás létezésére (Frobenius)
  + |A| = |B| és A bármilyen X részhalmazára |X| <= |N(X)|
  + BIZONYÍTÁS
    - A Hall tételből következik
    - Ha páros gráfban van teljes párosítás, akkor |A| = |B| triviális, és mivel legalább A-t le kell fedni, a második feltétel is igaz
    - Ha X <= |N(X)|, akkor létezik A-t lefedő párosítás és mivel |A| = |B| így az lefedi B-t is
* Ekvivalencia páros gráfban A-t lefogó párosítás létezésére (Hall)
  + A bármilyen X részhalmazára |X| <= |N(X)|
  + BIZONYÍTÁS
    - Ha van A-t lefedő párosítás, akkor a feltétel igaz
      * A minden eleméhez van hozzárendelve B legalább 1 „unique” eleme
    - Ha igaz a feltétel, akkor van ilyen párosítás
      * Azt akarjuk belátni, hogy ha a feltétel igaz, akkor v(G) >= |A|
        + Vagyis, hogy a gráfban több független él van, mint ahány csúcs van A-ban (ez páros gráfban ekvivalens azzal, hogy van A-t lefedő párosítás)
      * Legyen U egy minimális lefogó ponthalmaz, melynek Ua része van A-ban és Ub része van B-ben (|U| = T(G))
      * Ha X-nek A \ Ua-t választjuk, akkor tudjuk, hogy az X-ből jövő éleket mind Ub fedi le, vagyis |N(X)| <= |Ub|
      * Ekkor tudjuk, hogy
        + v(G) = T(G) = |U| = |Ua|+|Ub| >= |Ua|+|N(X)| >= |Ua|+|X| = |A|
        + Vagyis v(G) >= |A|
* Ekvivalencia teljes párosítás létezésére (Tutte)
  + V valamilyen X részhalmazára Cp(G – X) <= |X|
  + BIZONYÍTÁS (ha van párosítás, akkor az állítás igaz)
    - Ha van teljes párosítás, akkor X él elhagyásával minden leszakadt páratlan komponensből ment legalább egy teljes párosítás-beli él X-be, ráadásul X-nek mindig különböző tagjába

# Folyamok és összefüggőség

## Definíciók

* Hálózat
  + Olyan irányított gráf, melyben minden élre értelmezett egy élekről nemnegatív valósokra képző c(e) függvény, illetve ki van benne jelölve egy s és t különböző pont (termelő és fogyasztó)
* Folyam
  + Egy élekről nemnegatív egészekre képző függvény, mely egy gráf minden élén értelmezett és
    - 0 <= f(e) <= c(e) (vagyis legfeljebb a kapacitást rendeli hozzá)
    - És sum( f(uv) ) = sum( f(vu) ), vagyis érvényes rá Kirchoff csomóponti törvénye s és t-n kívül minden csúcsára
* Folyam értéke
  + Egy folyamban s-ből kifolyó nettó folyammennyiség
* S-T vágás
  + Legyen X egy s-et tartalmazó és t-t nem tartalmazó csúcshalmaz. Ekkor X-ből G\X-be mutató élek halmazát nevezzük S-T vágásnak
* Vágás kapacitása
  + Egy S-T vágás élein lévő kapacitások összege
* Éldiszjunkt utak
  + P és Q éldiszjunkt utak, ha kezdő és végpontjuk megegyezik, de nincs közös élük
* Pontdiszjunkt utak
  + P és Q pontdiszjunkt utak, ha végpontjaik megegyeznek, minden más pontjuk pedig eltér
* Egy gráf U pont / él halmaza lefog minden UV utat, hogyha elhagyásukkal nincs út u-ból v-be
* k-szorosan összefüggő gráfok
  + Bárhogy hagyok el belőlük k-1 pontot, összefüggő marad, de van k olyan pont, melyet ha elhagyok már nem összefüggő
* k-szorosan élösszefüggő
  + Bárhogy hagyok el belőlük k-1 élet, összefüggő marad, de van k olyan él, melyet ha elhagyok már nem összefüggő

## Tételek

* Egy folyam értéke akkor és csak akkor maximális, ha nincs javítóút s-ből t-be
  + BIZONYÍTÁS
    - Ha egy folyam értéke maximális akkor értelemszerűen nem javítható
    - Ha nincs javítóút, attól még vannak olyan pontok (halmazuk legyen X) melyekbe el lehet jutni javítóúttal (X-ben benne van s és G\X-ben benne van t)
    - Ekkor ha veszünk egy X-ből G\X-be mutató élet, azon f(e) = c(e) különben a G\X-ben lévő pont X-ben lenne
    - Ha veszünk egy G\X-ből X-be mutató élet, azon ugyanezen ok miatt f(e) = 0
    - Vagyis találtunk egy telített vágást, így a folyam maximális
* Ford-Fulkerson tétel
  + Létezik olyan vágás, melynek értéke a maximális folyam értéke (ez a minimális vágás)
    - Biztos, hogy nincs ennél kisebb vágás, ha lenne, az azt jelentené, hogy a vágáson több víz át tud folyni, mint a vágás kapacitása
    - Az előző tételben meg találtunk egy módszert amivel kereshető egy ilyen vágás
* EgÉr (egészértékűségű lemma)
  + Ha egy hálózatban minden kapacitás egész szám, akkor létezik olyam maximális folyam amin minden folyamérték egész szám
* Edmonds-Karp tétel
  + A javítóutas algoritmus (ha mindig a legrövidebb javítóutat választjuk) polinom idő alatt leáll
* Általánosítások
  + Több forrás és fogyasztó esetén felveszünk egy s’ forrást, melyből óriási kapacitású éleket vezetünk a valódi forrásra, fogyasztóval ugyanígy
  + Ha a pontoknak is van C kapacitásuk, akkor minden ilyen pontot helyettesítünk két kapacitás nélküli ponttal, melyek között épp egy C értékű él megy
  + Ha vannak irányítatlan élek, akkor felveszünk ugyanolyan kapacitással mindkét irányba mutató élt
* Menger 1
  + Irányított gráfban az s-ből t-be menő éldiszjunkt utak maximális száma megegyezik a lefogó utak minimális számával
  + BIZONYÍTÁS
    - Az biztos, hogy ha az éldiszjunkt utak száma k, akkor a lefogó utak száma >= k
    - Legyen a most a minimális lefogó utak száma k és minden él kapacitása 1
    - Ekkor a minimális vágás értéke legalább k, vagyis a maximális folyam értéke is legalább k
    - Vegyünk egy s-t utat (ilyen biztosan van, hiszen van k értékű folyam) és változtassuk az éleinek kapacitását 0-ra
      * A max folyam érték alsó határa már csak k-1
    - Ezt megtehetjük k-szor, és ezzel biztosan találtunk kdb éldiszjunkt utat
* Menger 2
  + Irányított gráfban az s-ből t-be menő pontdiszjunkt utak maximális száma megegyezik a lefogó pontok minimális számával
  + BIZONYÍTÁS
    - Szedjünk szét x != s, t minden pontot xbe és xki komponensre, és mutasson közöttük 1db él
    - Alkalmazzuk erre a gráfra a Menger 1-et, legyen K az eredeti gráfban a pontdiszjunkt utak száma
    - Ebben a Menger 1 szerint adódik K lefogó út, tehát K különböző éle, amelyek minden s-t utat lefognak
    - Ezeknek az éleknek van egy s-t-től különböző végpontja, melyek szerint megfeleltethetők az eredeti gráf egyik pontjának
* Menger 3
  + Irányítatlan gráfban az s-ből t-be menő éldiszjunkt utak maximális száma megegyezik a lefogó utak minimális számával
  + BIZONYÍTÁS
    - Helyettesítsünk minden élet egy odavissza irányított éllel (G’)
    - Alkalmazzuk a Menger 1-et, legyen K az erdeti gráfban az éldiszjunkt utak maximális száma
    - Ekkor G’-ben is K ilyen út van, mert a behozott duplaélek közül, találunk K olyat ami mindig csak az egyiket használja
      * Ha lenne 2 út amik közül az P1 az xy, P2 az yx élet használná, akkor P1 átírható úgy, hogy x pontba érve P2 x utáni részén megy tovább, P2 pedig ugyanígy y-nal
    - Ezzel tudjuk, hogy G’-ben K lesz a minimális lefogó élek száma, emiatt G-ben maximum K lehet
* Menger 4
  + Irányított gráfban az s-ből t-be menő pontdiszjunkt utak maximális száma megegyezik a lefogó pontok minimális számával
  + BIZONYÍTÁS
    - Visszavezetjük Menger 3-ra és 2-re
* Egy gráf akkor és csak akkor k-szorosan (él)összefüggő, ha bármely két pontja (/éle) között létezik kdb pont(/él)diszjunkt út
  + BIZONYÍTÁS (élekre)
    - Ha k-szorosan élösszefüggő, akkor tetszőleges 2 pontja közt nyilván legalább k lefogó útja van, vagyis (Menger 3) legalább k éldiszjunkt útja van
    - Ha bármely két pontja közt k éldiszjunkt útja van, akkor k él elhagyásával nem eshet k-nál több darabra, vagyis k-élösszefüggő
  + BIZONYÍTÁS (pontokra)
    - Ha k-összefüggő, akkor bármely 2 pontja között legalább k pontra van szükség a lefogáshoz, vagyis (Menger 4) legalább ennyi pontidegen útja van
    - Ha k pontdiszjunkt útja van bármely 2 pontja közt, akkor akárhogy hagyok el k-nál kevesebb pontot, összefüggő marad
* Menger 5
  + Legalább 3 pontú gráfra: pontosan akkor 2-összefüggő, ha bármely 2 pontján vezet át kör, és akkor 2-szeresen élösszefüggő, ha bármely 2 élén vezet át kör
  + BIZONYÍTÁS
    - Az első állítás triviálisan igaz, hiszen 2 pontdiszjunkt út egy kör
    - A második esetben, választunk tetszőleges e f éleket, kettéválasztjuk őket egy Pe és Pf ponttal, majd ezekre alkalmazzuk az első tételt
* Ha egy gráf aciklikus, akkor van benne nyelő
  + Felhasználjuk: minden aciklikus gráfban kell, hogy legyen topologikus sorrend
  + Vegyük a gráfban a leghosszabb irányított utat, tegyük fel, hogy a végpontja nem nyelő
  + Ekkor, ha topologikus sorrendben megyünk végig ezen az úton, akkor ez a pont vagy nem az utolsó, vagy mutat belőle visszafele él
* Egy gráf pontosan akkor aciklikus, ha van benne topologikus sorrend
  + Keressünk a gráfban legalább egy nyelőt és dobjuk ki
  + A fennmaradó gráfban is keressünk és dobjuk ki
  + Ha már minden csúcsot kidobtunk, akkor a kidobás fordított sorrendje egy topologikus sorrend

# Algoritmusok

* Javító utas algoritmus (teljes párosítás keresése)
  + Találtunk egy M párosítást tetszőleges gráfban
  + Veszünk egy u pontot amelyet nem fed le a párosítás és elmegyünk u szomszédjaiba
    - u szomszédjait biztos lefedi M, hiszen ha nem fedné le, csak egyszerűen bővíthetnénk M-et ezzel az éllel
  + Most azon az élen megyünk ezeknek a szomszédjaiba, melyek M-ben vannak
  + Ezekből a pontokból azokon az éleken megyünk tovább, melyek nincsenek M-ben
    - Tehát felváltva haladunk M-beli és nem M-beli éleken (alternáló út)
  + Ha eljutottunk egy pontba ami nincs benne M-ben, akkor a kapott alternáló út minden élét ami M-ben volt kivesszük M-ből és mindegyiket ami nem volt benne, azt beletesszük
    - Ezzel az éleink még mindig függetlenek, és mivel M-en kívüli éllel kezdtünk és végeztünk, egyel megnöveltük M elemszámát
  + Ha ezt minden u-ra megtesszük, és nem találtunk javítóutat, akkor a kapott út maximális
* Javító utas algoritmus (maximális folyam keresése)
  + Legyen adott egy G hálózat és benne egy folyam
  + Vegyük fel azt a hálózatot, melyben benne van az összes G-beli él, és azok visszafele mutató párjai, feltéve, hogy kapacitásaik pozitívak
  + Ebben a hálózatban az eredeti élek kapacitása legyen c(e) – f(e), a visszafele mutatóké f(e)
  + Ha most található folyam s-ből t-be, akkor találtunk egy javítóutat, a javított folyamban az élek folyamértéke f1(e) + f2(e), ha az él iránya megegyezik a G-beli irányával és f1(e) – f2(e), ha nem
* BFS (szélességi bejárás): lineáris futásidejű
  + Választunk egy tetszőleges csúcsot, felvesszük az összes szomszédját és azoknak azon szomszédait melyeket nem vettünk fel
  + Ezzel bejártuk a teljes gráfot, sőt, egy feszítőfáját (BFS-fa) is megkaptuk
  + TÉTEL: Ha s-ből t-be vezet irányított út a BFS fában, akkor aza legrövidebb irányított út (/utak egyike)
* Kruskal algoritmus (minimális összsúlyú feszítőfa)
  + Adott egy gráf és egy minden élre értelmezett élsúly függvény
  + Rendezzük az éleket élsúly szerinti növekvő sorrendbe
  + Járjuk be az élet, ha nem alkot kört
  + Végeztünk, felvettük az összes pontot, vagy az élsorozat végére jutottunk
  + TÉTEL: Kruskallal megkereshetjük a minimális összsúlyú feszítőfát
    - BIZONYÍTÁS
    - Belátjuk, hogy Kruskallal tényleg feszítőfát kaptunk
    - Belátjuk, hogy a kapott feszítőfa ideális (teljes indukció)
    - F0-ra triviális
    - Tfh Fi-1-re ideális
    - Ekkor a Kruskal hozzáfeszi az aktuális legkisebb súlyú élet. Ha ezzel Fi nem lenne ideális, az azt jelentené, hogy nem ezt az élet kellett volna bevenni (a fa többi részével nem lehet baj, hiszen Fi-1 ideális), vagyis, hogy bevehettünk volna egy kisebb súlyú élet.
      * Ez lehetetlen, ha lett volna ilyen él, azt a Kruskal megtalálta volna
* Dijkstra algoritmusa (legrövidebb út keresése): n^2 futásidejű
  + Adott egy G irányított, nemnegatív valósokkal súlyozott gráf
  + Olyan élsorozatot keresünk s-ből, melyen a súlyok összege a legkisebb
  + Számon tartjuk minden s-től különböző pontra, hogy eddig mekkora a legrövidebb út (d(v)) s-ből odajutni (kezdeti hossza minden ilyen pontnak végtelen)
  + Számon tartjuk, hogy melyik pontokból próbálkoztunk már javítással (T) és melyikekből nem (S)
  + Az első pont amiből próbálkozunk az s, utána S-ek közül mindig azzal, akinek ahova a legrövidebb eljutni
    - Az eredeti Dijkstra minden lépésben minden S-beli ponttal próbálkozott, ez viszott n^3 futásidejű volt
  + Az általános lépés, hogy a vizsgált (u) pontból kimenő minden uv élre megnézzük, hogy d(u) + f(uv) nem kisebb-e d(v). Ha az, akkor átjavítjuk
  + Végül átrakjuk a vizsgált u pontot S-ből T-be
  + Akkor végeztünk, ha már minden ponttal próbálkoztunk (|S| = 0)
* Ford algoritmusa (legrövidebb út negatív élsúlyok engedélyezésével): e\*n futásidejű (rosszabb mint n^2)
  + Adott egy G irányított valós számokkal súlyozott gráf
  + Az algoritmus megállapítja a gráfról, hogy konzervatív-e (nincs-e benne negatív kör), és ha igen, akkor egy s pontból az összes legrövidebb utat megadja
  + Rögzíteni kell egy tetszőleges élsorrendet
  + Minden lépésben minden él mentén megpróbálunk javítani a megadott sorrendben
  + És ezeket addig csináljuk újra és újra, amíg
    - Már nem történik javítás
    - Több javítást csináltunk, mint ahány csúcs van (ilyenkor van negatív kör)
* Floyd algoritmusa (minden 2 pont között megkeresi a legrövidebb utat): n^3
  + Adott egy G irányított valós számokkal súlyozott konzervatív gráf
  + Számozzuk be a pontokat, és minden 2 pont között értelmezzünk egy súlyfüggvényt, úgy, hogy ha a két pont között nincs él, akkor a köztük lévő él súlya végtelen, egyébként pedig az eredeti súlyukat (vagy önmaguk esetén 0 súlyú hurokélet) kapják
  + Legyen d1, d2… dn+1, vagyis dk(i, j) azon „hosszfüggvény”, mely megadja i és j közötti legrövidebb utat k-nál kisebb indexű pontokon keresztül, így dn+1 megadja a legrövidebb utat
  + k = 2-vel kezdünk, d1-nek beállítjuk a súlyfüggvényt
  + Az általános lépés, hogy minden csúcspárra ellenőrizzük, hogy a k. indexű csúcs javít-e a jelenleg hozzá rendelt hosszon, vagyis d(k)(i, j) nagyobb-e d(k)(i, k) + d(k - 1)(k, j)-nél, és ha igen, akkor javítunk
  + Ezt ismételgetjük k = n + 1-ig
* Leghosszabb / (legrövidebb) út keresése DAG-ban
  + Vegyük a gráf nyelőit, és rakjuk őket egy halmazba, és dobjuk ki a gráfból
  + A maradék gráfon végezzük ezt el ugyanígy és tegyük rendezzük ezt a halmazt az előzőtől balra
  + Most balról jobbra menjünk végig a halmazokon és mindegyik eleméhez rendeljünk egy t(u) időt az alábbi módon
    - Ha korábbi halmazok összes olyan vi elemére, hogy létezik vi-u (ei) él
    - t(u) = max( t(vi) + t + w(e) )
  + Más szóval topologikus sorrendben bejárjuk a csúcsait és minden csúcshoz az őt megelőző csúcsokból a leghosszabb útút választjuk
* DFS algoritmus
  + Számon tartjuk minden csúcs mélységi (megtalálási) és befejezési (elhagyási) számát
  + Számon tartunk ezen kívül még egy stacket, melybe a csúcsokat tesszük megtalálási sorrendben és dobjuk ki a legfelsőt elhagyási sorrendben. Futás közben mindig a stack legfelső elemét vizsgáljuk
  + Az algoritmus kiválasztja az s csúcsot, stackbe teszi, és jelzi, és megjegyzi, hogy őt találta meg először (mélységi száma 1)
  + Ha van olyan még meg nem talált csúcs amibe a stack tetejéről tudunk menni, akkor „megtaláljuk”, ő kapja a következő mélységi számot és ő kerül a stack tetejére
  + Ha nem lehet ilyen csúcsba tovább menni, akkor stack tetején lévő elem megkapja a következő befejezési számot, és kikerül a stack tetejéről és abból a csúcsból folytatjuk a vizsgálatot, amelyikből a stack tetején lévőből jutottunk, vagyis amelyik a kidobással a stack tetején maradt
  + Ha a stack kiürült, akkor végeztünk („befejeztük” az összes csúcsot)
  + TÉTEL
    - G pontosan akkor aciklikus, ha egy DFS futása során nem keletkezett visszaél.
    - Ekkor a topologikus sorrend a befejezési sorrend fordítva
    - BIZONYÍTÁS
      * Ha van visszaél, az állítás triviális
      * Tegyük fel tehát, hogy nincs és lássuk, be, hogy ha f(v) a befejezési számokat rendeli hozzá, akkor minden uv élre f(v) < f(u)
        + Ha uv előreél vagy faél, akkor értelemszerűen u-t előbb találtuk meg, mint v-t és u-t még nem fejeztük be, amikor v-hez értünk, így v-t előbb fogjuk befejezni
        + Ha uv keresztél, az azt jelenti, hogy u-t akkor találtuk meg, amikor v már be is volt fejezve, így f(u) nyilván nagyobb lesz f(v)-nél