Figyelem! *Ez a dokumentum a készítéskor a bme.ysolt.net oldalon megtalálható, számomra ismeretlen szerzőjű „szamtud-vizsga-tetelek.doc" file átirata. Módosításaim célja elsősorban ezen tételsor aktualizálása a 2009/2010/1. félév szerinti tematikához, másodsorban pedig néhány megfogalmazás egyértelműsítése. Az is egyértelmű továbbá, hogy a számomra ismeretlen szerző munkája készítésekor a Katona–Recski–Szabó–féle Számítástudomány alapjai jegyzetből átvett megfogalmazásokat. Ezen okok miatt leszögezem, hogy* ***ezt a dokumentumot alapvetően nem én írtam, de nem célom senki munkájának a sajátomként feltüntetése!*** *Kérem a munka értékelésekor ezt a szempontot mindig figyelembe venni! Eredményes vizsgát kívánok!*

Budapest, 2010. január 25.

Kondor Máté András
BME–VIK, 2009/V07. tankör

A számítástudomány alapjai, BME – VIK, 2009/10/1
Vizsgatematika

1. Leszámlálási alapfogalmak (permutációk, variációk és kombinációk, ismétlés nélkül, vagy ismétléssel), binomiális tétel, szita–formula

Permutáció:

- **ismétlés nélkül**: *n* elem összes lehetséges sorrendje: $n! = \prod_{i=0}^{n-1}(n-i)$
  - megjegyzés: $0! \triangleq 1$
- **ismétléssel**: k_1 darab első típusú elem, k_2 darab második elem, ..., k_n darab *n*. típusú elem lehetséges sorba állításai: $\frac{(k_1+k_2+\ldots+k_n)!}{k_1!\cdot k_2!\cdot\ldots\cdot k_n!} = \frac{\left(\sum_{i=1}^{n} k_i\right)!}{\prod_{i=1}^{n}(k_i!)}$

Kombináció:

- **ismétlés nélkül**: egy *n* elemű halmaz *k* elemű részhalmazai: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
- **ismétléssel**: *n* elemből *k* kiválasztása, ha a sorrend nem számít, de az elemek többször is szerepelhetnek: $\binom{n+k-1}{k}$

Variáció:

- **ismétlés nélkül**: *n* elemből az összes lehetséges sorrendben *k* darab különböző kiválasztása: $\frac{n!}{(n-k)!}$
- **ismétléssel**: *n* elemből képezhető *k* tagú sorozatok (egy–egy elem többször is szerepelhet): n^k

A Newton–féle binomiális tétel:

- szeretnénk $(a+b)^n$ polinom alakját megadni, ehhez tekintsük a következő példát:
  $(a+b)^3 = (a+b)(a+b)(a+b) = aaa + aab + aba + baa + abb + bab + bba + bbb =$
  $= aaa + aab + aab + aab + abb + abb + abb + bbb = a^3 + 3a^2b + 3ab^2 + b^3$
- figyeljük meg, hogy jönnek létre az utóbbi összeg tagjai: minden zárójelből szerepel egy tag
- általában, egy $(a+b)^n$ alakú kifejezés kifejtésében $a^k b^{n-k}$ típusú tagok szerepelnek, ahol a kitevők összege éppen *n*-nel egyenlő
- mivel *n* zárójelből *k* számút $\binom{n}{k}$-féleképpen választhatunk ki, az eredményben $\binom{n}{k}$-szor szerepel az $a^k b^{n-k}$ tag, ez azt jelenti, hogy a kéttagú *n*. hatványa $\binom{n}{k} a^k b^{n-k}$ alakú tagok összegéből áll, ahol *k* értéke 0-tól *n*-ig terjedhet, ezért
  $(a+b)^n = \binom{n}{0}a^n + \binom{n}{1}a^{n-1}b + \ldots + \binom{n}{n-1}ab^{n-1} + \binom{n}{n}b^n = \sum_{i=0}^{n}\left[\binom{n}{i}a^{n-i}b^i\right]$

– binomiális együtthatók néhány nevezetes tulajdonsága:
  – $\binom{n}{k} = \binom{n}{n-k}$
  – $\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$
  – $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{n-1} + \binom{n}{n} = \sum_{i=0}^{n} \binom{n}{i} = 2^n$ (azaz az n elemű halmaznak 2^n részhalmaza van)
  – $\binom{n}{0} = \binom{n}{n} = 1$
  – $\binom{n}{k} = 0, \text{ ha } k > n$

A szita–formula

– adott a következő probléma: mondjuk meg, hány 1000-nél kisebb természetes szám van, mely nem osztható sem 2-vel, sem 3-mal, sem 5-tel
– a megoldás a szita–módszerrel történik: összesen 999 db. szám közül kereshetjük a feltételt teljesítőket, ebből le kell vonni a 2-vel, 3-mal és 5-tel oszthatókat, de így kétszer is levontuk a $2 \cdot 3$-mal, a $3 \cdot 5$-tel és a $2 \cdot 5$-tel oszthatókat, így ezeket hozzáadjuk, így viszont háromszor levontuk, de háromszor hozzá is adtuk a $2 \cdot 3 \cdot 5$-tel oszthatókat, így ezeket megint levonjuk, tehát: $999 - \left\lfloor \frac{999}{2} \right\rfloor - \left\lfloor \frac{999}{3} \right\rfloor - \left\lfloor \frac{999}{5} \right\rfloor + \left\lfloor \frac{999}{2 \cdot 3} \right\rfloor + \left\lfloor \frac{999}{3 \cdot 5} \right\rfloor + \left\lfloor \frac{999}{2 \cdot 5} \right\rfloor - \left\lfloor \frac{999}{2 \cdot 3 \cdot 5} \right\rfloor = 266$
– ugyanezzel a módszerrel adható meg például egy tetszőleges n-hez relatív prímek száma, vagy különböző halmazok uniójának elemszáma

2. Gráfelméleti alapfogalmak, fák egyszerűbb tulajdonságai, Kruskal tétele (minimális költségű feszítőfa keresése), Cayley tétele (fák száma), Prüfer–kód

Gráfok alapvető tulajdonságai

- egy **gráf** egy **rendezett pár**, $G=(V,E)$, ahol *V (vertex)* egy nem üres halmaz, *E (edge)* pedig ebből a halmazból képezhető párok egy halmaza
- *V* elemeit **pontok**nak vagy **csúcsok**nak, *E* elemeit **élek**nek nevezzük
- ha egy *G* gráfról beszélünk, akkor $V(G)$-vel illetve $E(G)$-vel jelöljük a gráf pontjainak illetve éleinek halmazát, míg a pontok illetve élek számát $v(G)$-vel illetve $e(G)$-vel jelöljük
- ha az $e\in E$ él a $\{v_1,v_2\}$ párnak felel meg, akkor ez a két pont *e* végpontjai
- ha $v_1=v_2$ akkor *e* **hurokél**
- ha két különböző élnek a végpontjai azonosak, a két élet **párhuzamos**, vagy **többszörös él**nek nevezzük
- azokat a gráfokat, amelyekben nincsenek hurokélek és többszörös élek, **egyszerű gráf**nak nevezzük
- ha $e,f\in E$ végpontjai $\{v_1,v_2\}$ illetve $\{w_1,w_2\}$, és van közös végpontjuk, tehát $\{v_1,v_2\}\cap\{w_1,w_2\}\neq 0$, akkor *e, f* **szomszédos élek** és hasonlóan, v_1 és v_2 **szomszédos pontok**, ha egy élre illeszkednek, tehát $\{v_1,v_2\}\in E$
- v_1 illeszkedik *e*-re, ha annak egyik végpontja
- egy pont **izolált pont**, ha nem illeszkedik egyetlen élre sem
- egy *v* pontra illeszkedő élek száma a pont **fokszáma**, jelölése $d(v)$
- a maximális fokszámot Δ-val, a minimálist δ-val fogjuk jelölni
- egy esetleges hurokél kettővel növeli a fokszámot
- ***k*–reguláris** egy gráf, ha minden pontjának foka *k*
- ha egy *n* pontú egyszerű gráf tetszőleges két pontja szomszédos, akkor ***n*–pontú teljes gráf**nak nevezzük, és K_n-el jelöljük
- a $G=(V,E)$ és a $G'=(V',E')$ gráfok **izomorf**ak, ha van olyan egy–egy értelmű megfeleltetés *V* és *V'* között, hogy *G*-ben pontosan akkor szomszédos két pont, ha *G'*-ben a nekik megfelelő pontok szomszédosak és szomszédos pontpárok esetén ugyanannyi él fut köztük
- a $G'=(V',E')$ gráf a $G=(V,E)$ gráf **részgráf**ja, ha $V'\subseteq V$, $E'\subseteq E$, valamint egy pont és egy él pontosan akkor illeszkedik egymásra *G'*-ben, ha *G*-ben is illeszkedők
- a *G'* **részgráf komplementere** az a $G''=(V'',E'')$ gráf, amiben az eredeti gráf minden pontja és a részgráfból „kimaradt” élek vannak, tehát $V''=V$ és $E''=E-E'$
- egy *G* **gráf komplementere** alatt azt a $\overline{G}$ gráfot értjük, amelyet akkor kapunk, ha *G*-t a $K_{v(G)}$ teljes gráf részgráfjának tekintjük, vagyis $\overline{G}$-ben azok a pontpárok vannak összekötve, amelyek *G*-ben nincsenek
- ha *E'* azokból az *E*-beli élekből áll, amelyeknek mindkét végpontja *V'*-ben van, és *E'* az összes ilyen élet tartalmazza, akkor *G'* a *G* gráf *V'* által **feszített részgráf**ja

- egy $(v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k)$ sorozatot **élsorozat**nak nevezünk, ha e_i a v_{i-1}-t és v_i-t összekötő él, ha $v_0 = v_k$, akkor az élsorozat **zárt**, ha a csúcsok mind különbözőek, akkor ez egy **út**, ha pedig $v_0 = v_k$ és a csúcsok mind különbözőek, akkor ez egy **kör** a gráfban
- az út, illetve kör hosszán az őt alkotó élek számát értjük
- definiáljuk a $p \equiv q$ relációt úgy, hogy $p \equiv q \Leftrightarrow p, q \in V(G)$ és vezet út *p* és *q* között, vagy $p = q$, ez egy **ekvivalencia reláció**
- a fenti reláció **ekvivalenciaosztályok**at határoz meg G pontjain
- az egy osztályba eső pontok által feszített részgráfokat a G gráf összefüggő **komponens**einek hívjuk, számukat $c(G)$-vel jelöljük
- ha a komponensek száma 1, vagyis ha *G* bármely két pontja között vezet út, akkor a *G* gráf **összefüggő**
- egy $X \subseteq E$ élhalmazt **elvágó élhalmaz**nak nevezünk, ha az *X*-beli élek elhagyásával nő a gráf komponenseinek száma, azaz a gráf több komponensre esik, mint ahányból eredetileg állt
- *X* **vágás**, ha az legkisebb elvágó élhalmaz, tehát, ha elvágó, de semelyik valódi részhalmaza sem
- az egyelemű vágásokat **elvágó él**eknek nevezzük

Irányított gráfok

- élei nem $\{v_1, v_2\}$ alakú rendezetlen, hanem (v_1, v_2) alakú rendezett párok, egy ilyen (v_1, v_2) élnek v_1 a kezdőpontja, v_2 a végpontja, rajzban az élet egy v_1-ből v_2-be mutató nyíllal ábrázoljuk
- **forrás**nak hívunk egy pontot, ha egyetlen élnek sem végpontja, **nyelő**nek, ha egyetlen élnek sem kezdőpontja
- irányított gráfban egy $(v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$ utat akkor hívunk **irányított út**nak, ha $e_1 = (v_0, v_1),\ e_2 = (v_1, v_2),\ e_k = (v_{k-1}, v_k)$, az **irányított kör** definíciója hasonló
- egy irányított gráf **erősen összefüggő**, ha bármely pontjából bármely más pontjába vezet irányított út

Általános összefüggések

- minden gráfban a fokszámok összege az élek számának kétszerese, tehát $\sum_{v_i \in V(G)} \left[d(v_i) \right] = 2\left[e(G) \right]$
- minden gráfban páros a páratlan fokú pontok száma
- a legalább két pontot tartalmazó egyszerű gráfnak van két azonos fokú pontja
- az *n* pontú teljes gráf éleinek száma: $e(K_n) = \frac{n(n-1)}{2}$

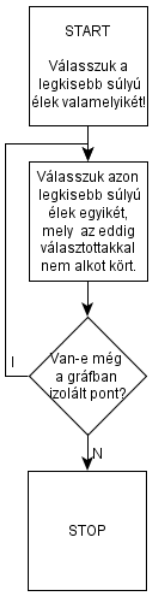
Fák és tulajdonságaik:

- az összefüggő, körmentes gráfokat **fák**nak nevezzük
- minden legalább 2 pontú fában van legalább két elsőfokú pont
- egy *n* pontú fa éleinek száma $n-1$
- az *F* gráf a *G* gráf **feszítőfája**, ha *F* fa, pontjainak halmaza megegyezik *G* pontjainak halmazával, tehát $V(F) = V(G)$ és *F* élei szerepelnek G-ben is
- minden összefüggő gráf tartalmaz feszítőfát
- a körmentes gráfokat **erdő**nek nevezzük, ezek nem feltétlenül egykomponensűek

– egy F a G gráf **feszítőerdő**je, ha F erdő, és minden komponense feszítő fája G megfelelő komponensének

Kruskal–algoritmus

– animáció: http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/kruskal/Kruskal.shtml
– egy algoritmust **mohó algoritmus**nak nevezünk, ha végrehajtása folyamán minden lépésben az éppen a legjobbnak tűnő lehetőséget választjuk, és nem törődünk azzal, hogy esetleg egy most rosszabbnak tűnő választással végül jobb eredményt kaphatnánk
– a **Kruskal–algoritmus** egy mohó algoritmus a legkisebb súlyú feszítőerdő megkeresésére, a mohó algoritmus azonban más feladatok, például a legkisebb súlyú kör megkeresésére vagy páros gráfban a maximális párosítás megkeresése esetén nem feltétlenül ad jó megoldást
– rendeljünk egy gráf éleihez **súly**okat, nem negatív valós számokat
– jelöljük $s(e)$-vel az e-hez rendelt súlyt
– ha $X \subseteq E(G)$, akkor X súlya $s(X) = \sum_{e \in X} \left[s(e) \right]$
– az algoritmus, amely megkeresi a minimális súlyú feszítőerdőt G-ben, **Kruskal algoritmusa**:
  (1) válasszuk ki a gráfból a legkisebb súlyú élek közül bármelyiket
  (2) válasszuk ki a legkisebb súlyú élek közül bármelyiket, amelyik nem alkot kört az eddig már kiválasztottakkal
  (3) ha ilyen nincs, tehát a gráf minden pontjára illeszkedik kiválasztott él, megállunk, ha van, akkor ezt a (2)-est ismételjük



– bizonyítsuk, hogy az előbbi algoritmus G minimális súlyú feszítőerdejét adja
  – nyilvánvaló, hogy az algoritmus végén a kiválasztott élek egy F feszítőerdőt alkotnak
  – tegyük fel indirekt, hogy F_0 minimális súlyú feszítőerdő, és $s(F_0) < s(F)$
  – ha több ilyen ellenpélda van, akkor ezek közül azt válasszuk F_0-nak, amelynek a lehető legtöbb közös éle van F-el
  – legyen $e_0 \in E(F_0) - E(F)$
  – ha e_0-t hozzávesszük F-hez, akkor kapunk egy C kört
  – ha valamely $e \in E(C) - \{e_0\}$ élre $s(e) > s(e_0)$ állna, akkor az algoritmus során e helyett e_0-t választottunk volna, így $s(e) \le s(e_0)$ igaz minden $e \in E(C)$-re
  – mivel $F_0 - e_0$ legalább két komponensből áll, van legalább egy olyan $e_1 \in E(C) - \{e_0\}$ él, amely összeköti $F_0 - e_0$ két komponensét
  – nyilván feszítőerdő $F_1 = (F_0 - e_0) \cup \{e_1\}$ is
  – már láttuk, hogy $s(e) \le s(e_0)$, nem lehet azonban $s(e_1) < s(e_0)$, mert ekkor $s(F)_1 < s(F_0)$ igaz volna, ami ellentmond F_0 minimalitásának
  – csak $s(e_1) = s(e_0)$ lehetne, de ekkor F_1 olyan ellenpélda lenne, amelynek eggyel több közös éle van F-el, mint F_0-nak, ez pedig ellentmond a feltevésnek

A Prüfer–kód

– az $\{1, 2, \ldots, n\}$ pontokon adott fához rendeljünk egy számsorozatot a következőképpen:
  – számozzuk meg a fa pontjait tetszőleges sorrendben

  - töröljük le a fa elsőfokú pontjai közül a legkisebb indexűt, és jegyezzük fel a szomszédja (a vele összekötött egyetlen pont) indexét, legyen ez v_1
  - ismételjük az eljárást a maradék fára egészen addig, amíg csak egy pont marad
  - világos, hogy az utolsó pont az *n* sorszámú
  - ezt biztosan nem hagytuk el soha, hiszen mindig legalább két elsőfokú pont volt, és nyilván nem lehetett a legkisebb sorszáma *n*, ezért nem is kell, hogy a számsorozat végén feltüntessük
- az így kapott $v_1, v_2, \ldots, v_{n-2}$ sorozatot a fa **Prüfer–kód**jának nevezzük

Cayley tétele

- lássuk be Cayley tételét, mely szerint: az $\{1, 2, \ldots, n\}$ pontokon n^{n-2} különböző fa adható meg
  - egy fához nem tartozhat két különböző Prüfer–kód, és minden fához tartozik Prüfer–kód
  - azt kell még belátnunk, hogy minden sorozathoz tartozik egy fa, amelynek a Prüfer–kódja épp az adott sorozat
  - abból, hogy hány számból áll a Prüfer–kód, könnyen meghatározhatjuk v_{n-1}-et, hiszen az biztosan *n*-el egyenlő
  - legyen w_k az a pont, amelyik elhagyásánál v_k-t feljegyeztük, elég tehát meghatározni w_k-t, minden *k*-ra, ebből már egyértelműen rekonstruálható a fa
  - w_1 a legkisebb szám, ami nem fordul elő a Prüfer–kódban, pontosabban $v_1, v_2, \ldots, v_{n-1}$ között
  - általában w_k pedig a legkisebb szám, ami nem fordul elő a $w_1, w_2, \ldots, w_{k-1}, v_k, v_{k+1}, \ldots, v_{n-1}$ számok között
  - mivel ez legfeljebb $n-1$ darab különböző szám, mindig van ilyen legkisebb szám
  - így megkaptuk, hogy a fa élei $\{v_1, w_1\}, \{v_2, w_2\}, \ldots, \{v_{n-1}, w_{n-1}\}$, ezek az élek tényleg fát határoznak meg, ez abból látható, hogy ha nem fa lenne, akkor létezne benne kör, viszont abban a lépésben, amikor létrejönne a kör, egy olyan csúcsot írtunk volna fel, ami már szerepelt, ez azonban a fenti módszernél nem fordulhat elő
  - könnyen látható, hogy ennek a fának Prüfer–kódja éppen $v_1, v_2, \ldots, v_{n-1}$
  - tehát minden olyan $n-1$ elemű sorozathoz, amelyben az első $n-2$ elem mindegyike lehet $\{1, 2, \ldots, n\}$, és az utolsó elem *n*, tartozik egy-egy fa, és különböző sorozathoz különböző fa tartozik
  - mivel ilyen sorozat n^{n-2} van, ennyi a különböző fák száma is

3. Euler–séta és –körséta (Euler–út, –kör) fogalma, szükséges és elégséges feltétel a létezésükre, Hamilton–kör és –út, szükséges, illetve, elégséges feltételek Hamilton-kör létezésére

Euler–út és –kör

- a *G* gráf **Euler-út**jának nevezünk egy élsorozatot, amely pontosan egyszer tartalmazza *G* összes élét
- ha az élsorozat zárt, akkor **Euler–kör**t kapunk
- lássuk be a tételt, mely szerint egy *G* gráfban akkor és csak akkor van Euler–kör, ha *G* minden pontjának fokszáma páros, és *G* összefüggő
  - először lássuk be, hogy ha van a gráfban Euler–kör, akkor minden pont foka páros
    - induljunk el a gráf egy tetszőleges pontjából, és járjuk körbe az Euler–kör vonalán, így nyilvánvalóan minden pontba pontosan annyiszor „mentünk be”, ahányszor „kimentünk”, de a „kimenések” és a „bemenések” számának összege épp a pont fokszáma, ez pedig így biztosan páros
  - hogy *G*-nek összefüggőnek kell lennie az nyilvánvaló
  - a másik irányt *G* pontszámára való indukcióval bizonyítjuk
    - tegyük fel, hogy minden $k < n$ -re igaz az állítás, és legyen *G* egy *n* pontú gráf
    - induljunk el a gráf egy tetszőleges pontjából, és haladjunk az élek mentén úgy, hogy egy élen kétszer nem megyünk át
    - ha egy olyan pontba érünk, amelyből nem vezet ki olyan él, amelyen még nem haladtunk át, akkor ez csak a kiinduló pont lehet, mivel minden pont foka páros, így tehát egy zárt élsorozatot kapunk
    - legyen a *H* egy olyan zárt élsorozata *G*-nek, amelyben az előforduló élek száma maximális
    - mivel a kiindulópontból már nem tudtunk tovább menni, az ebből a pontból kiinduló minden él *H*-beli
    - indirekt tegyük fel, hogy *H* nem egy Euler–köre *G*-nek
    - vizsgáljuk a *G'* gráfot, amelyet úgy kaptunk, hogy a *G* gráfból elhagytuk a *H*-ban szereplő éleket
    - *G'* nem feltétlenül összefüggő, viszont összesen *n*-nél kevesebb pontja van, hiszen a kiindulópont nincs benne
    - az indukciós feltevés miatt minden komponensében van Euler–kör
    - mivel *G* összefüggő, *G'* valamelyik komponensének van olyan pontja, amelyik *H*-ban szerepel
    - nevezzük az ebben a komponensben található Euler–kört *H'*-nek
    - tehát ha elindulunk az előbb talált közös pontból és először bejárjuk *H*-t majd *H'*-t, akkor egy *H* élszámánál nagyobb élszámú zárt élsorozatot találtunk, ami ellentmond a feltevésünknek, vagyis *H* Euler-kör
- ugyanígy bebizonyítható, hogy egy gráfban akkor és csak akkor van Euler-út, ha a páratlan fokú pontok száma 0 vagy 2 és a gráf összefüggő
- fontos megjegyezni, hogy nevük ellenére az Euler–körök és –utak nem a gráfelmélet alapfogalmai szerint értelmezett körök és utak egy gráfban ellentétben a most következő Hamilton–körrel és –úttal, melyek tényleges körök és utak

Hamilton–út és –kör

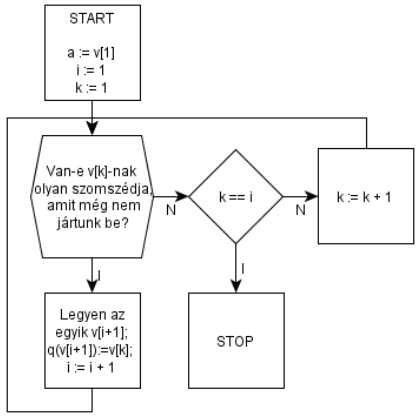
- egy gráfban **Hamilton–körnek** nevezünk egy kört, ha *G* minden pontját pontosan egyszer tartalmazza
- egy utat pedig **Hamilton–útnak**, ha G minden pontját pontosan egyszer tartalmazza
- ha *G*-ben van *k* olyan pont, melyeket elhagyva a gráf több mint *k* komponensre esik szét, akkor nem létezik a gráfban Hamilton–kör
- ha *G*-ben van *k* olyan pont, melyeket elhagyva a gráf több mint $k+1$ komponensre esik szét, akkor nem létezik a gráfban Hamilton–út
- a Hamilton–kör létezésére több elégséges feltétel született
- bizonyítsuk **Ore tételét**, amely szerint, ha az *n* pontú G gráfban nincs olyan $x,y \in V(G)$ amelyre $d(x)+d(y)<n$ és $\{x,y\} \notin E(G)$, vagyis nincs szomszédos pontpár, melyre igaz lenne, hogy fokszámaik összege kevesebb, mint a gráf pontjainak a száma, akkor *G*-ben van Hamilton–kör
  - indirekt tegyük fel, hogy a gráf kielégíti a feltételt, de nincsen benne Hamilton–kör
  - vegyünk hozzá a gráfhoz éleket úgy, hogy továbbra se legyen benne Hamilton–kör
  - ezt egészen addig csináljuk, amikor már akárhogyan is veszünk hozzá egy élet, lesz a gráfban Hamilton–kör
  - az így kapott *G'* gráfra továbbra is teljesül a feltétel
  - biztosan van két olyan pont, hogy $\{x,y\} \notin E(G')$
  - ekkor a $G'+\{x,y\}$ gráfban van egy Hamilton–kör, tehát *G'*-ben van Hamilton–út
  - legyen ez $P=(z_1,z_2,\ldots,z_n)$, ahol $z_1=x$ és $z_n=y$
  - legyenek $z_{i_1},z_{i_2},\ldots,z_{i_k}$ az *x* pont szomszédai a *P* úton $(2=i_1<i_2<i_3<\ldots<i_k<n)$
  - ekkor *y* nem lehet összekötve z_{i_a-1}-el $(1\le a\le k)$, mert $(z_1,\ldots,z_{i_a-1},z_n,z_{n-1},\ldots,z_{i_a})$ egy Hamilton–kört adna
  - így tehát $d(y)\le n-1-d(x)$, ami viszont ellentmondás, mert $\{x,y\} \notin E(G)$
- bizonyítsuk **Dirac tételét**, amely szerint, ha egy *n* pontú *G* gráfban minden pont foka legalább $\frac{n}{2}$, akkor a gráfban létezik Hamilton–kör
  - ez az előző tételből következik, hiszen ha minden pont foka legalább $\frac{n}{2}$, akkor teljesül az Ore–tétel feltétele, mivel nincs olyan $\{x,y\}$ amelyre $d(x)+d(y)<n$

4. Legrövidebb utakat kereső algoritmusok (BFS, Dijkstra, Ford, Floyd)

- a következő eljárások során mindig adott egy összefüggő $G=(V,E)$ gráf és egy kitüntetett $s\in V$ pontja
- a legrövidebb út megkeresésére különböző eljárások vannak:

BFS (Breadth–First–Search, szélességi keresés)

- animáció: http://www.cs.bme.hu/~gsala/alg_anims/3/graph2-e.html
- ha az élek száma adja az út hosszát, azaz az élek nem súlyozottak, a szélességi keresést (BFS) kell alkalmazni
- a szélességi bejárás lényege, hogy a kezdőpontból először elmegyünk annak összes szomszédjába, majd az első szomszéd összes olyan szomszédjába, ahol még nem jártunk
- ha az összes szomszédot bejártuk, elmegyünk abba a pontba, ahol a legrégebben jártunk, és szomszédain folytatjuk a bejárást, mindezt addig folytatjuk, ameddig tudjuk
- ha *s* és *t* közti legrövidebb utat keressük, a kiválasztott *t* ponthoz meghatározzuk a $q(t)=u_1$ pontot
  - ha $u_1\neq s$, akkor a $q(u_1)=u_2$ pontot, és így tovább, amíg $u_k=s$ lesz, és akkor az *s*-ből *t*-be vezető legrövidebb út az $(s=u_k,u_{k-1},\ldots,u_1,t)$ lesz



- a bejárás lépésszáma $(e+v)$-vel arányos, valamint még annyi lépés kell, amilyen hosszú a keresett út

A Dijkstra–algoritmus

- animáció: http://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html
- az élek különböző hosszúságú utakat modelleznek, azaz súlyozottak
- ha bejárjuk a gráfot, az *u* pontra $t(u)$ értéke azt fogja jelenteni, hogy az algoritmus során eddig talált, az *s* kezdőpontból *u*-ba vezető utak közül a legrövidebb hossza $t(u)$, $l(e)$ jelenti *e* él hosszát
- a **Dijkstra–algoritmus** csak akkor használható, ha semelyik él hossza sem negatív
- az algoritmus kulcslépése a következő helyettesítés: ha *x*-ből vezet egy *e* él *y*-ba és $d(y)>d(x)+l(e)$, akkor $d(y)\leftarrow d(x)+l(e)$
- a Dijkstra–algoritmus szerint minden él mentén a javítást csak egyszer kell elvégezni, ha már biztosak vagyunk abban, hogy az *e* él kezdőpontjának *t* értéke utána már nem fog tovább csökkenni
  (1) lépés: $S\leftarrow\{s\};\ T\leftarrow V-\{s\};\ t(s)=0;\ \forall u\neq s\text{-re } t(u)\leftarrow\infty$
  (2) lépés: javítás u_0-ból a *T*-beli pontokba vezető *e* élekre
  (3) lépés: *T*-beli pontok közül legyen u_0 az, amelyiken a $t(u)$ érték a legkisebb, tegyük át u_0-t *T*-ből *S*-be
  (4) lépés: ha *T* üres, az algoritmus befejeződött, különben újra a (2) lépéstől
- az algoritmus lépésszáma legfeljebb v^2-tel arányos

A Ford–algoritmus

- animáció: http://links.math.rpi.edu/applets/appindex/graphtheory.html
- ha olyan irányított gráfot tekintünk, melyben az élhosszak között lehetnek negatív számok, de bármely irányított kör mentén az élhosszak összege nem negatív kell, hogy legyen, **Ford algoritmus**át használjuk a legrövidebb út keresésére
  (1) lépés: számozzuk meg az éleket 1-től *e*-ig, legyen $i \leftarrow 1$, és $t(s) = 0;\ \forall u \neq s$-re $t(u) \leftarrow \infty$
  (2) lépés: a rögzített sorrendben végezzük el a Dijkstra–algoritmusnál említett javítást minden élen
  (3) lépés: $i \leftarrow i+1$, ha $i > v$, akkor az algoritmus befejeződött, különben újra a (2) lépéstől
- az algoritmus lépésszáma $e \cdot v$-vel arányos

A Floyd–algoritmus

- animáció: http://links.math.rpi.edu/applets/appindex/graphtheory.html
- a **Floyd–algoritmus**sal az összes pontpár távolsága meghatározható, ehhez egy olyan súlyozott gráf kell, melyben nincs negatív összsúlyú irányított kör
  (1) lépés: $\forall(i,j)$ rendezett párra legyen $t^{(1)}(i,j) \leftarrow l(i,j)$, továbbá $k \leftarrow 1$
  (2) lépés: $\forall(i,j)$ rendezett párra $t^{(k+1)}(i,j) \leftarrow \min\left\{t^{(k)}(i,j); t^{(k)}(i,k) + t^{(k)}(k,j)\right\}$
  (3) lépés: ha $k = n+1$, akkor az algoritmus befejeződött, különben $k \leftarrow k+1$ és újra a (2) lépéstől
- az algoritmus lépésszáma v^3-el arányos

5. Párosítások, König–Hall–tétel, Frobenius–tétel

Párosítások

- egy *G* gráfot páros gráfnak nevezünk, ha a *G* pontjainak $V(G)$ halmaza két részre, egy *A* és egy *B* halmazra osztható úgy, hogy *G* minden élének egyik végpontja *A*-ban, másik végpontja *B*-ben van
- a $K_{a,b}$ -vel jelölt teljes páros gráf olyan $G=(A,B)$ páros gráf, ahol $|A|=a$ és $|B|=b$, és amelyben minden *A*-beli pont össze van kötve minden *B*-beli ponttal
- egy *G* gráf akkor és csak akkor páros gráf, ha minden *G*-ben levő kör páros hosszúságú
- egy párosítás teljes, ha a gráf minden pontját lefedi

A Hall–tétel

- egy $G=(A,B)$ páros gráfban akkor és csak akkor van *A*-t lefedő párosítás, ha *A* ponthalmaz minden valós részhalmazára igaz, hogy annak pontjainak száma kisebb, vagy egyenlő, mint annak szomszédos pontjainak száma (ez a **Hall-feltétel**), formálisan: $(\forall X \subseteq A)(|N(X)| \geq |X|)$, ahol $|N(X)|$ az *X* halmazbeli pontokkal szomszédos pontok száma
- ezen tétel egyszerű következménye a következő tétel

A Frobenius–tétel

- egy $G=(A,B)$ páros gráfban akkor és csak akkor van teljes párosítás, ha $|A|=|B|$ és igaz a Hall–feltétel *A* minden valós részhalmazára

6. König és Gallai tételei, Tutte–tétel (bizonyítás nélkül)

Új jelölések
- $\nu(G)$: a G gráfban található független élek maximális száma (nincs közös végpontjuk)
- $\tau(G)$: a lefogó pontok minimális száma (minden él egyik végpontját tartalmazzák)
- $\rho(G)$: a lefogó élek minimális száma (a lefogó élhalmaz minden pontot lefog)
- $\alpha(G)$: független pontok maximális száma (független ponthalmazban nincs szomszédos pont)

I. tétel
- bizonyítsuk a tételt, mely szerint minden gráfra $\nu(G) \le \tau(G)$
  - legyen *M* egy maximális méretű független élhalmaz, *X* pedig egy minimális méretű lefogó ponthalmaz
  - mivel *M* különböző éleit csak különböző pontokkal lehet lefogni, $\nu(G) = |M| \le |X| = \tau(G)$

II. tétel
- bizonyítsuk a tételt, mely szerint minden gráfra $\alpha(G) \le \rho(G)$
  - a bizonyítás menete olyan, mint az I. tétel esetében

III. (Gallai) tétel
- bizonyítsuk a tételt, mely szerint $\tau(G) + \alpha(G) = v(G)$ minden hurokmentes gráfra
  - egy *X* halmaz pontjai akkor és csak akkor függetlenek, ha a $V(G) - X$ halmaz lefogó ponthalmaz
    - tehát $\tau(G) \le |V(G) - X|$ teljesül minden *X* független ponthalmazra
    - ebből pedig következik, hogy $\tau(G) + \alpha(G) \le v(G)$
    - hasonlóképpen $\alpha(G) \ge |V(G) - Y|$, minden *Y* lefogó ponthalmazra amiből $\tau(G) + \alpha(G) \ge v(G)$ következik

IV. (Gallai) tétel
- bizonyítsuk a tételt, mely szerint $\nu(G) + \rho(G) = v(G)$ minden izolált pont nélküli gráfra
  - egy $\nu(G)$ elemű *X* független élhalmaz lefog $2 \cdot \nu(G)$ különböző pontot
  - a többi pont lefogható $v(G) - 2 \cdot \nu(G)$ éllel, így $v(G) - \nu(G) > \rho(G)$
  - másrészt, ha *Y* egy minimális lefogó élhalmaz, akkor *Y* *k* darab diszjunkt csillag egyesítése, ha ugyanis *Y* tartalmazna 3 hosszú utat, akkor a középső élet el lehetne hagyni
  - így $\rho(G) = v(G) - k$
  - ha minden csillagból kiválasztunk egy élet, az így kapott élhalmaz nyilván független
  - tehát $\nu(G) \ge k = v(G) - \rho(G)$

V. (König) tétel

- bizonyítsuk a tételt, mely szerint páros gráfnál $\nu(G)=\tau(G)$, ha nincs izolált pontja a gráfnak, $\alpha(G)=\rho(G)$ is teljesül
  - legyen *M* egy maximális párosítás, azaz $|M|=\nu(G)$
  - legyenek *X* ill. *X'* az *M* által lefogott *A* illetve *B*-beli pontok halmazai
  - legyen $U=A-X$, *T'* azon *B*-beli pontok halmaza, amelyek elérhetők *U*-ból alternáló úton, *T* pedig ezek párjainak halmaza; $T'\subseteq X'$
  - legyen $Y=T'\cup(X-T)$
  - ennek a halmaznak éppen $\nu(G)$ pontja van
  - ezek minden élet lefognak, mert $N(T'\cup U)=T'$. Így $\tau(G)\le Y$, amiből az I. tétel – $\nu(G)\le\tau(G)$ – alapján következik az állítás
  - a másik állítás Gallai két tételéből és König tételének első részéből következik

A Tutte–tétel

- tetszőleges gráfra általánosít: egy *G* gráfban akkor és csak akkor létezik teljes párosítás, ha minden $X\subseteq V(G)$-re $c_p(G-X)\le|X|$, azaz akárhogy hagyunk el a gráfból néhány pontot, a maradékban a páratlan komponensek száma ennél több nem lehet ($c_p(H)$ jelöli a *H* gráf páratlan sok pontot tartalmazó komponenseinek számát)

7. Hálózati folyamok, Ford–Fulkerson–tétel, Edmonds–Karp–tétel (bizonyítás nélkül)

- legyen *G* egy irányított gráf
- rendeljünk minden éléhez egy $c(e)$ nemnegatív valós számot, amit az él kapacitásának nevezünk
- jelöljünk ki továbbá két (*s*, *t*) pontot *G*-ben, melyeket termelőnek (forrásnak) illetve fogyasztónak (nyelőnek) hívunk
- ekkor a (G,s,t,c) négyest hálózatnak nevezzük
- az él kapacitásán egy megengedett függvényt (átáramló mennyiséget) folyamnak hívunk
- ha a nyelőbe beérkező összmennyiséget növelni akarjuk, találhatunk ún. javító utakat, melyek segítségével egyes folyamértékeket növelve, másokat esetleg csökkentve az össz folyamérték nő
- egy folyam értéke akkor és csak akkor maximális, ha nincs javító út *s*-ből *t*-be
- legyen $s \in X \subseteq V(G) - \{t\}$, így nyilvánvalóan $t \in V(G) - X$ és $s \notin V(G) - X$
- azoknak az éleknek a *C* halmazát, amelyeknek egyik végpontja *X*-beli, másik $\left[V(G) - X\right]$-beli, a hálózati folyam egy (*s*, *t*)-vágásának nevezzük
- a vágás értéke, $c(C)$, azon az éleken levő kapacitások összege, amelyek egy *X*-beli pontból egy $\left[V(G) - X\right]$-beli pontba mutatnak
- a **Ford–Fulkerson–tétel** kimondja, hogy a maximális folyam értéke egyenlő a minimális vágás értékével
- az **Edmonds–Karp–tétel** kimondja, hogy, ha mindig a legrövidebb javító utat vesszük, akkor a maximális folyam meghatározásához szükséges lépések száma felülről becsülhető a pontok számának polinomjával

8. Menger tételei, gráfok összefüggőségi számai, Dirac–tétel (bizonyítás nélkül)

– két út élidegen egymástól, ha nincs közös élük, hasonlóképpen értelmezünk pontidegen utakat is

Menger tételei

(1) ha *G* egy irányított gráf, $s,t \in V(G)$, akkor az *s*-ből *t*-be vezető páronként élidegen irányított utak maximális száma megegyezik az irányított *s*-ből *t*-be vezető utakat lefogó élek maximális számával

(2) ha *G* egy irányított gráf, $s,t \in V(G)$ két nem szomszédos pont, akkor az *s*-ből *t*-be vezető, végpontoktól eltekintve pontidegen irányított utak maximális száma megegyezik az irányított *s*-ből *t*-be vezető utakat *s* és *t* felhasználása nélkül lefogó pontok minimális számával

(3) ha *G* egy irányítatlan gráf, $s,t \in V(G)$, akkor az *s*-ből *t*-be vezető élidegen irányítatlan utak maximális száma megegyezik az irányítatlan *s*-ből *t*-be vezető utakat lefogó élek minimális számával

(4) ha *G* egy irányítatlan gráf, $s,t \in V(G)$ két nem szomszédos pont, akkor az *s*-ből *t*-be vezető pontidegen irányítatlan utak maximális száma megegyezik az irányítatlan *s*-ből *t*-be vezető utakat *s* és *t* felhasználása nélkül lefogó pontok minimális számával

Gráfok összefüggőségei

– egy G gráfot *k*-szorosan **összefüggő**nek nevezünk, ha legalább $k+1$ pontja van és akárhogy hagyunk el belőle *k*-nál kevesebb pontot, a maradék gráf összefüggő marad
– a gráf *k*-szorosan **élösszefüggő**, ha akárhogy hagyunk el belőle *k*-nál kevesebb élet, összefüggő gráfot kapunk
– egy *G* gráf akkor és csak akkor *k*-szorosan élösszefüggő, ha legalább $k+1$ pontja van, és bármely két pontja közt létezik *k* pontidegen út
– hasonlóan *G* akkor és csak akkor *k*-szorosan élösszefüggő, ha bármely két pontja között létezik *k* élidegen út
– a *G* gráf akkor és csak akkor kétszeresen összefüggő, ha tetszőleges két ponton át vezet kör
– igaz az is, hogy akkor és csak akkor kétszeresen összefüggő, ha bármely két élen át vezet kör
– **Dirac tétele** kimondja, hogy, ha a *G* gráf *k*-szorosan összefüggő, akkor bármely $x_1, x_2, \ldots, x_k$ pontján át vezet kör

9. Pont– és élszínezés, korlátok a kromatikus számra, Mycielsky–konstrukció, Brooks–tétel (bizonyítás nélkül)

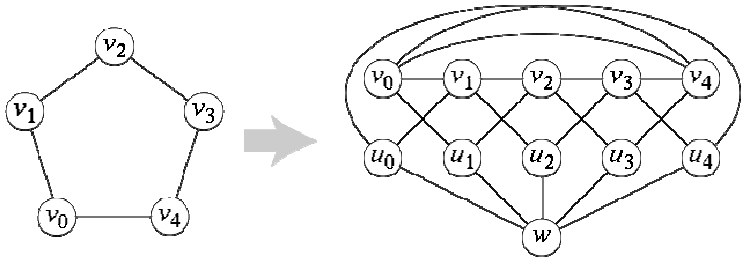
- egy *G* hurokmentes gráf ***k* színnel kiszínezhető**, ha minden csúcsot ki lehet színezni legfeljebb *k* színnel úgy, hogy bármely két szomszédos csúcs színe különböző legyen.
- *G* kromatikus száma $\chi(G)=k$, ha *G* *k* színnel kiszínezhető, de $k-1$ színnel nem
- egy ilyen színezésnél az azonos színeket kapott pontok halmazát **színosztály**nak nevezzük
- egy *G* gráf egy teljes részgráfját **klikk**nek nevezzük, és a *G*-ben található összes klikk közül a legnagyobb adja a gráf **klikkszám**át, melynek jelölése $\omega(G)$

Korlátok a kromatikus számra

- teljes gráfokra $\chi(G)=v(G)$, nem teljes gráfokra $\chi(G)\le v(G)$
- minden *G* gráf esetében igaz, hogy a kromatikus szám nagyobb, vagy egyenlő, mint a klikkszám, vagyis $\chi(G)\ge\omega(G)$
- ez az alsó korlát a kromatikus számra éles például olykor, ha a gráf egy teljes gráf, de van olyan gráf is, amire nagyon rossz ez a korlát

Mycielsky konstrukciója

- a **Mycielski–konstrukció** a $V(G)=\{v_1,\ldots,v_n\}$ csúcshalmazú *G* gráfhoz egy olyan gráfot rendel, melyben feszített részgráfként szerepel a *G* gráf, továbbá még $n+1$ csúcs, a következő elrendezésben:
  - minden *G*-beli v_i csúcsnak van egy u_i párja, melynek szomszédsága megegyezik a v_i szomszédságával, vagyis azokkal és csak azokkal a csúcsokkal van összekötve, amelyekkel v_i
  - az $(n+1)$. új csúcs (*w*) mindegyik u_i csúccsal össze van kötve, de egyetlen v_i-vel sincs.
- **Mycielski–gráf**oknak azokat a gráfokat nevezzük, amelyek a kétpontú teljes gráfból, vagyis a két pontból és egyetlen élből álló gráfból előállíthatóak a fenti eljárás egymás után következő véges számú alkalmazásával.



- legyen *G* olyan gráf, melyre igaz, hogy $\chi(G)=k$, ekkor **Mycielski tétele** szerint minden $k\ge 2$ egész számra létezik olyan *G* gráf, hogy $\omega(G)=2$ és $\chi(G)=k$, vagyis a kromatikus szám felső korlátja és a klikkszám nem függenek egymástól

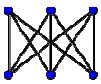
Brooks–tétel

- ha *G* egyszerű, összefüggő, nem teljes gráf, és nem egy páratlan hosszúságú kör, akkor $\chi(G)\le\Delta=\max_{x\in V(G)}\lceil d(x)\rceil$, azaz ekkor a kromatikus szám nem nagyobb, mint a maximális fokszám

10. Síkbarajzolhatóság, Euler–féle poliédertétel, Kuratowski tétele (csak könnyű irányban bizonyítani), Fáry–Wagner–tétel (bizonyítás nélkül)

- ha egy gráf lerajzolható a síkba úgy, hogy az élei ne messék egymást, akkor az **síkbarajzolható**
- a síkbarajzolt gráf a síkot tartományokra osztja
- hasonlóan definiáljuk a **gömbre rajzolható** gráfot
- egy *G* gráf pontosan akkor síkbarajzolható, ha gömbre rajzolható
  - tegyük a gömböt a síkra
  - ahol a gömb a síkkal érintkezik, nevezzük déli pólusnak
  - az északi pólusból egyeneseket kell bocsátani a gráf pontjaira
  - az egyenes metszi a gömb felületét, ezek a pontok felelnek meg a sík pontjainak
  - ez az úgynevezett **sztereografikus projekció**
  - az eljárás megfordítható, ha az északi póluson nincs pont és nem halad át rajta é
- egy összefüggő síkbeli gráf, amelynek *n* csúcsa, *e* éle és *t* tartománya van (beleértve a külső, nem korlátos tartományt is), eleget tesz az **Euler–formulának**: $n-e+t=2$
  - tekintsünk egy gráf *C* körét és ennek egy *a* élét
  - a *C* kör a síkot két tartományra osztja, ezeket egyéb élek további tartományokra osztanak, de mindig van egy olyan melynek *a* a határa
  - ha *a*-t elhagyjuk, a két tartomány egyesül, azaz a tartományok száma 1-el csökken
  - a csúcsok száma nem változik, tehát *a* elhagyásával az $n-e+t$ érték nem változik
  - ezt az eljárást folytatva végül egy fát kapunk
  - elég erre belátni, ez viszont triviális, hiszen $t=1$ és minden fára $e=n-1$
- ha G egyszerű, síkbarajzolható gráf és pontjainak száma 3, akkor az előbbi jelöléssekkel $e \le 3n-6$
- az ábrákon látható **Kuratowski–gráfok** $\left(K_{3,3} \text{ és } K_5\right)$ nem síkbarajzolhatók

  - $K_{3,3}$ esetében a pontok száma 6, az élek száma 9, így az Euler-formulából $t=5$
  - minden egyes tartomány határának legalább négy élet kell tartalmaznia, hiszen ha volna olyan, amelynek a határa három élből állna, akkor ezek között lenne két ház, vagy két út, amelyek össze vannak kötve, ez pedig ellentmondás

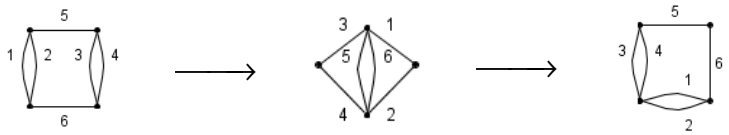


  - minden él két tartománynak a határán van, így teljesül $4t \le 2e$, de ez a $9 \ge 10$ ellentmondáshoz vezet
  - tehát $K_{3,3}$ nem síkbeli
  - K_5 esetén minden tartomány határa legalább három élet tartalmaz, vagyis $3t \le 2e$
  - mivel azonban az Euler–formulából $t=7$ és $e=10$, ismét ellentmondásra jutottunk
- egy gráf síkbarajzolhatóságát nem befolyásolja, ha egy élet 2 hosszú úttal helyettesítünk, azaz egy élet egy új 2 fokú csúcs felvételével két élre bontunk, vagy ha egy 2 fokú csúcsra illeszkedő éleket egybeolvasztjuk
- a *G* és *H* gráfok **topológikusan izomorfak**, ha a fent említett transzformációk ismételt alkalmazásával izomorf gráfokba transzformáljuk őket
- **Kuratowski tétele** szerint egy gráf akkor és csak akkor síkbarajzolható, ha nem tartalmaz olyan részgráfot, amely topológikusan izomorf $K_{3,3}$-mal vagy K_5-tel
- a **Fáry–Wágner–tétel** szerint, ha *G* egy egyszerű, síkbarajzolható gráf, akkor létezik olyan síkbeli ábrázolása is, hogy minden élet egy egyenes szakasszal rajzolunk le

11. Dualitás, gyenge izomorfia, Whitney tételei (bizonyítás nélkül), síkgráfok színezése, ötszíntétel

Dualitás, gyenge izomorfia, Whitney tételei

- *G* tartományaihoz rendeljük az új *G'* gráf pontjait és *G'*-ben akkor kössünk össze két pontot éllel, ha a megfelelő két *G*-beli tartománynak van közös határvonala
- az így rajzolt gráfot *G* **duálisának** nevezzük
- ez a *G'* gráf síkbarajzolható lesz
- két gráf **gyengén izomorf**, ha élei között kölcsönösen egyértelmű és körtartó leképezés hozható létre köztük
- probléma: egy gráftól elvárjuk, hogy duálisának duálisa ismét az eredeti(vel izomorf) legyen
- nyilván ez nem teljesül, mert ha az ábra első gráfjának a duálisát valaki úgy rajzolja le, mint a következő gráf akkor az ismételt duálisképzés az utolsó gráfhoz fog vezetni



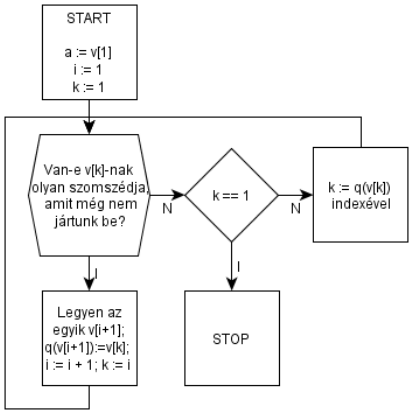
- a problémát a gyenge izomorfia segítségével oldja meg **Whitney tétele**: legyen *G* síkbarajzolható gráf és *H* vele gyengén izomorf, ekkor *H* is síkba rajzolható, *G'* és *H'* szintén gyengén izomorfak, végül *(G')'* és *(H')'* gyengén izomorfak *G*-vel, illetve *H*-val
- a gyenge izomorfiát élek közötti körtartó megfeleltetésnek definiáltuk, természetes ötletnek tűnik, hogy a dualitást is definiáljuk élek közötti, kört vágásba vivő megfeleltetésként
- az eddigi definícióktól való megkülönböztethetőség érdekében mondjuk azt, hogy a *G* és *G'* gráfok egymás **absztrakt duálisa**i, ha éleik közt létesíthető olyan, kölcsönösen egyértelmű leképezés, mely kört vágásba, vágást körbe visz
- nyilvánvaló, hogy ha *G* síkbarajzolható, akkor a lerajzolás után nyert „régi" duálisa egyben absztrakt is lesz
- megmutatható, hogy síkba nem rajzolható gráfhoz más eljárással sem lehet duálist rendelni, mivel **Whitney tétele szerint** egy gráfnak akkor és csak akkor létezik absztrakt duálisa, ha síkbarajzolható
- már rég észrevették, hogy ha egy politikai térképen a szomszédokat nem lehet azonos színre színezni, akkor a feladat megoldható 4 színnel
- mivel mi a pontokhoz rendelünk színeket, a térképet az alábbiak alapján átalakíthatjuk: *G* tartományaihoz (az országokhoz) rendeljük az új *G'* gráf pontjait és *G'*-ben akkor kössünk össze két pontot éllel, ha a megfelelő két *G*-beli tartománynak van közös határvonala
- az így gyártott gráfot szokás *G* duálisának is nevezni
- ez a *G'* gráf síkbarajzolható lesz
- **Whitney** harmadik **tétele** szerint egy *H* gráf akkor és csak akkor gyengén izomorf *G*-vel, ha *H*-ból az alábbi három lépés ismételgetésével *G*-vel izomorf gráfot kaphatunk:
  - ha *x* olyan pontja *H*-nak, hogy $H-\{x\}$ két komponensre esik, akkor „húzzuk szét" a gráfot két komponensre
  - az előző lépés fordítottja: ha *H* két komponensből áll, akkor a két komponensből válasszunk ki egy-egy pontot, és ezeknél fogva „ragasszuk össze" a kettőt
  - ha *x* és *y* olyan pontjai *H*-nak, hogy $H-\{x\}$ és $H-\{y\}$ is összefüggő, de $H-\{x,y\}$ már nem, akkor „húzzuk szét" a két részt, majd „ragasszuk össze" fordítva

Síkgráfok színezése

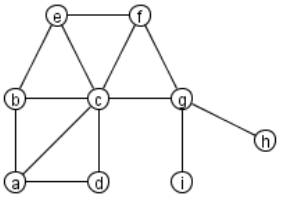
- bizonyítsuk az **ötszín–tétel**t, mely kimondja: ha *G* síkbarajzolható gráf, akkor $\chi(G) \leq 5$
  - a gráf pontszámára vonatkozó indukcióval bizonyítunk
  - mivel a párhuzamos élek nem befolyásolják a színezést, feltételezhetjük, hogy a gráf egyszerű
  - a *G* éleinek a száma legfeljebb $3n-6$ lehet (lásd fent)
  - így biztosan van egy olyan *x* pont, amelynek foka legfeljebb 5, hiszen ha minden pont foka legalább 6, akkor az élek száma legalább $\frac{1}{2}6n$ volna, ami ellentmondás
  - ha *x* foka legfeljebb négy, akkor az indukciós feltevés miatt *x*-et elhagyva kiszínezhető a gráf 5 színnel, majd *x*-et a 4 szomszédjától eltérő ötödik színnel színezzük ki
  - tegyük most fel, hogy $d(x)=5$
  - ha *x*-nél bármely két szomszédja között van él, akkor a gráfban egy K_6 részgráf szerepel, ami ellentmond *G* síkbarajzolhatóságának
  - tehát *x* két szomszédja, *y* és *z* nincs összekötve
  - húzzuk össze egy ponttá az *x*, *y* és *z* pontokat
  - az így kapott *G'* az indukciós feltevés miatt kiszínezhető 5 színnel
  - az ennek megfelelő színezés *G*-ben nem jó, hiszen *x*, *y*, *z* egyszínűek
  - *G*-ben *x*-nek három szomszédja van *y*-on és *z*-n kívül
  - ezek legfeljebb három színt foglalnak le, és a további két szomszéd, *y* és *z*, egyszínű
  - marad tehát az ötödik szín, amellyel kiszínezhetjük *x*-et
  - tehát *G* kiszínezhető 5 színnel

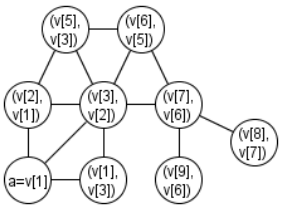
12. Mélységi keresés és alkalmazásai (pl. irányított kör létezésének eldöntése), PERT-módszer, kritikus tevékenységek

A mélységi keresés (DFS, Depth–First–Search)



- elindulunk a gráf egy pontjából és úgy járjuk be, hogy megnézzük, hogy a pontnak van-e még be nem járt szomszédja: ha van rálépünk, ha nincs akkor vissza az előző pontra
- ebből is látható, hogy két számot is rendelni kell a pontokhoz: az első: hányadiknak léptünk ide, a második: mi volt az előző
- ha már nincs több bejárandó pont, akkor automatikusan a kiindulási pontra jutunk vissza, ez a stop feltétel
- a folyamatábra mutatja a bejárási algoritmust: v_i jelöli az az *i.*-ként bejárt pontot, *k* annak a pontnak a sorszáma, melyről épp továbblépünk és $q(v_x)$ jelöli azt, ahonnan v_x-re léptünk
- az alább látható ábrákon látható egy gráf, valamint az, hogy mélységi bejárása során milyen v_i és $q(v_i)$ értékek adódtak





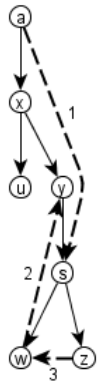
Alapkörrendszer keresése

- ha egy összefüggő *G* gráf valamely *F* fájához minden lehetséges módon hozzáveszünk még egy további $e \notin F$ élt, akkor $F \cup \{e\}$ egyetlen C_e kört tartalmaz
- ezen körök együttesét az *F* fához tartozó **alapkörrendszernek** vagy **fundamentális körrendszernek** vagy **báziskörrendszernek** nevezzük
- a mélységi keresés algoritmusa kis módosítással használható alapkörrendszer keresésére
- $y_0 := v_j$, majd $l = 1$-től kezdve $y_l := q(y_{l-1})$ egészen addig, míg v_i-ig nem jutunk, ekkor a mélységi keresés algoritmusában a stop helyén a következő kiegészítést kell tennünk: $(\forall e \notin F)\left[C_e = (y_0, y_1, \ldots y_k, y_0)\right]$

Irányított körök felismerése, emeletekre bontás

- induljunk ki egy tetszőleges a pontból és végezzünk mélységi bejárást
- ha nem jutottunk el minden ponthoz így, akkor válasszunk ki egy tetszőleges még be nem járt pontot és ismét végezzünk mélységi bejárást
- ezt az eljárást addig folytassuk, amíg minden pontot be nem jártunk
- az így kapott fák unióját nevezzük **DFS erdő**nek
- az ábrán látható módon három csoportba oszthatjuk egy irányított gráf éleit:

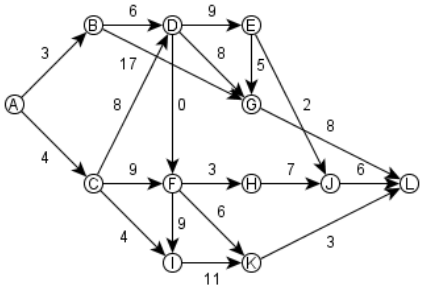
– a (p,p') irányított él lehet:
  (1) **előre–él**, ha *p* őse *p'*-nek (ekkor *p* sorszáma kisebb, mint *p'* sorszáma)
  (2) **vissza–él**, ha *p'* él őse *p*-nek (ekkor *p'* sorszáma kisebb)
  (3) **kereszt–él**, ha *p* és *p'* közül egyik sem őse a másiknak

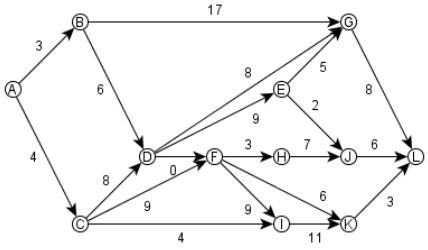


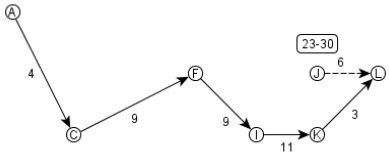
– pl. (a,s) él előre–él, (w,y) él vissza–él, a (z,w) él kereszt–él
– a mélységi erdőhöz tartozó élek minden előre–élek, a gráf többi éle bármilyen lehet, ha ugyanis a gráfban lenne pl. egy (u,y) él, akkor *u*-ból nem léptünk volna vissza *x*-be az *y* meglátogatása nélkül
– egy irányított gráfban akkor és csak akkor van irányított kör, ha a mélységi bejárás során találunk vissza–élt
– nevezzük **emeletekre bontás**nak az irányított gráf *V* ponthalmazának azt a $V = V_1 \cup V_2 \cup \ldots \cup V_m = \bigcup_{i=1}^{m} V_i$ partícióját (ha ilyen létezik), melyben (x,y) irányított élre, ha $x \in V_i$ és $y \in V_j$, akkor $i < j$ teljesül és nincs olyan él, ami két V_i-beli pont között fut
– ilyenkor V_1 elemei **források** (vagyis be-fokuk zérus) és V_m elemei **nyelők** (vagyis ki-fokuk zérus), de a források és nyelők létezése még nem garantálja az emeletekre bonthatóságot
– egy irányított gráfban akkor és csak akkor van irányított kör, ha ponthalmaza nem bontható emeletekre

A PERT (Program Evaluation and Review Technique) módszer

– az irányított gráf élei súlyozva vannak
– meg kell keresni a leghosszabb utat
– a gráfban nem lehet irányított kör
– az oldalt látható ábrán látható gráfon fogjuk magyarázni
– a felső ábra emeletekre bontható, melynek megvalósítását a középső mutatja
– balról jobbra haladva meghatározhatjuk minden tevékenység elkezdésének időpontját
– az eljárást a középső ábra szemlélteti: a bal szélső tevékenység azonnal megkezdhető, később egy *y* tevékenységhez tekintsük át az összes olyan $x_1, x_2, \ldots$ tevékenységet, melyre $(x_i,y) \in E(G)$ és ha ezek legkorábban a $t_1, t_2, \ldots$ időpontban kezdődnek el, akkor *y* elkezdése legkorábban a $\max\left\{\sum_{i=1}^{k}\left[t_i + l(x_i,y)\right]\right\}$ időpontban kerülhet sor
– a kritikus utat mutatja az alsó ábra







13. Keresési, beszúrási és rendezési algoritmusok (beszúrásos, buborék, összefésülés, láda), alsó korlátok a lépésszámokra, gráfok tárolása

Keresés

– keresünk egy *x* egész számot 1 és *n* között, erre a következő lehetőségeik vannak:
  (1) a **lineáris keresés**: sorban az összes számot kipróbáljuk az adott intervallumon, ekkor átlagosan $\frac{n}{2}$, legrosszabb esetben $n-1$ lépésre van szükségünk
  (2) a **logaritmikus keresés**: kipróbáljuk, hogy *x* nagyobb-e $\frac{n}{2}$-nél, így csak a sorozat felével kell foglalkoznunk, ezután újra megfelezzük a sorozatot, majd újra, stb. egész addig, amíg meg nem találjuk
– világos, hogy a logaritmikus keresés gazdaságosabb, ugyanis azzal a legrosszabb esetben is $\log_2 n$ lépésben megtaláljuk a megoldást
– természetesen az eljárások akkor is működnek, ha a sorozat nem szigorúan monoton növekszik vagy *n* nem épp 2 valamelyik hatványa, stb.
– összehasonlításképp egy kis táblázat a kétféle keresés hatékonyságáról:

n	(1) $n-1$	(2) $\lceil \log_2 n \rceil$
2	1	1
10	9	4
50	49	6
100	99	7
500	499	9
1000	999	10
100000	99999	17

– nem lehetséges olyan algoritmust készíteni, ami $\lceil \log_2 n \rceil$-nél kevesebb összehasonlítással, sőt általánosabban, ennél kevesebb bármilyen típusú eldöntendő kérdés feltevésével mindig megoldaná ezt a feladatot

Beszúrás

– *n* darab különböző számunk van növekvő sorrendben, és egy adott számot kell beszúrnunk
– ez a feladat ugyanaz, mintha *n* intervallumból kellene kiválasztanunk a megfelelőt, csak az *x* után következőket még el kell tolnunk egyel, így a legrosszabb esetben $n+\lceil \log_2 n \rceil$ lépést végzünk
– számos esetben az összehasonlítás lényegesen energiaforrásigényesebb művelet, mint a mozgatás

Sorba rendezés

– ezek után azonnal látszik, hogy *n* darab különböző valós szám sorba rendezéséhez sem kell több, mint $n \log_2 n$ darab összehasonlítás

- tekintsük ugyanis a számokat, amilyen sorrendben érkeznek, és mindig a legutoljára érkezettet szúrjuk be az addigiak közé
- világos, hogy összesen $a=\lceil \log_2 1\rceil+\lceil \log_2 2\rceil+\ldots+\lceil \log_2 n\rceil=\sum_{i=1}^{n}\lceil \log_2 i\rceil$ darab összehasonlítást végeztünk, ez körülbelül $n\log_2 n-n$ ami közel a legjobb érték
- most is elmondhatjuk, hogy amennyiben az összehasonlítások sokáig tartanak a beszúrásokhoz szükséges adatmozgatásokhoz képest, akkor ez közel optimális algoritmus
- az **összefésüléses rendezés** lényegében ugyanilyen lépésszámmal működik:
  - egy *k* és egy *l* hosszúságú rendezett tömb összefésülésekor összehasonlítjuk a két tömb legkisebb elemeit és közülük a kisebbet egy új tömbbe
  - utána a maradék két tömb két legkisebb eleme közül a kisebbet áttesszük az új tömb következő helyére
  - ezt a lépést ismételgetve $k+l-1$ lépés után rendezett tömböt kapunk
- további rendező algoritmusok:
  - az egyik különösen egyszerűen működik:
  - az adott *n* szám közül először kiválasztjuk a legkisebbet, majd a többi közül a legkisebbet stb.
  - az összehasonlítások száma nyilván $\frac{n(n-1)}{2}\approx cn^2$ és a további lépések száma is maximum ugyanennyi (hisz az összehasonlítás eredményétől függően vagy fel kell cserélni a két számot, vagy nem)
  - ha az összehasonlítások időigénye nem nagy, akkor egyszerűsége miatt gyakran használják ezt az algoritmust
- a **buborék rendezés** nem optimális, szintén cn^2 összehasonlítást használ, de egyszerű
  - páronként összehasonlítja és rendezi a tömb elemeit, így az első lépés végére a legnagyobb (vagy a legkisebb, ahogy beállítjuk) elem áll az utolsó helyen
  - a következő lépésben szintén páronként hasonlítgatjuk a tömböt, de az utolsó elemet már kihagyjuk a vizsgálatból
  - ezt a lépést ismételgetve rendezett tömbhöz jutunk
- a **láda rendezés** még $cn\log_2 n$-nél is gyorsabb, de csak akkor használható, ha a sorba rendezendő számok egészek és értékük pl.: 1-től *n*-ig terjed, csak ismeretlen sorrendben
  - nyissunk egy *n* hosszúságú $b(1),b(2),\ldots,b(n)$ tömböt
  - ha az *i*.-ként beérkező $a(i)$ szám értéke *j*, akkor a $b(j)$ helyre beírjuk az *i* értéket
  - ha ezt mind az *n* darab $a(i)$-re befejeztük, akkor a *b* tömbben épp az áll, hogy milyen sorrendben kell az $a(i)$ elemeket olvasnunk, hogy növekvő sorrendbe kerüljenek
  - ha pl. $n=5$ és $a=(3,1,5,2,4)$, akkor $b=(2,4,1,5,3)$ és csakugyan, az a tömb 2. eleme a legkisebb, a 4. eleme a következő, stb.

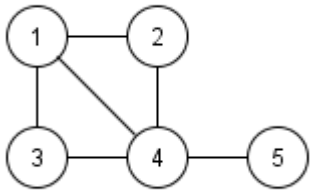
Gráfok tárolása

- definiáljuk az *n* pontú, *e* élű *G* gráfnak az $n\times e$ méretű $B(G)=(b_{ij})$ **illeszkedési mátrix**át oly módon, hogy $(b_{ij})=\begin{cases}0, & \text{ha a } j.\text{ él nem illeszkedik az } i.\text{ ponthoz}\\ 1, & \text{ha a } j.\text{ élnek az } i.\text{ pont kezdőpontja}\\ -1, & \text{ha a } j.\text{ élnek az } i.\text{ pont a végpontja}\end{cases}$
- megállapodás szerint $(b_{ij})=1$ akkor is, ha a *j*. él az *i*. ponthoz illeszkedő hurokél

- irányítatlan esetben is ez a definíció, csak ott a *j*. él mindkét végpontjának megfelelő elem 1
- definiáljuk a *G* gráf $n \times n$-es $A(G) = (a_{ij})$ **szomszédsági mátrix**át a következő módon:

$$(a_{ij}) = \begin{cases} 0, & \text{ha az } i. \text{ és } j. \text{ pont nem szomszédos} \\ k, & \text{ha az } i. \text{ és } j. \text{ pont között } k \text{ db párhuzamos él halad} \\ l, & \text{ha } i = j \text{ és az } i. \text{ ponthoz } l \text{ db hurokél illeszkedik} \end{cases}$$

- irányított gráfokat is megadhatunk ily módon, csak ott (a_{ij}) az *i*. pontból a *j*. pontba vezető élek száma
- **szomszédossági tömb**ről beszélünk, ha egy gráf csúcsainak száma mellé felírjuk a szomszédos csúcsok számát
- **rendezett szomszédossági tömb**ről, ha az egyes pontok szomszédai rendezett sorrendben szerepelnek
- a **láncolt szomszédossági lista** egy bonyolultabb adatszerkezet
  - először is, mint a szomszédossági tömb esetében, felírjuk az egyes csúcsok szomszédainak sorszámát egymás mellé, tetszőleges sorrendben
  - ezek után felírjuk egy másik sorozatba, hogy az első lépésben létrehozott sorozatban melyik csúcs szomszédai hányadik helyen kezdődnek
  - majd ezután, az első lépésben létrehozott sorozat minden tagja alá odaírjuk, hogy az először létrehozott sorozat hányadik helyén kell folytatni a kiolvasást
  - ha nem kell folytatni a kiolvasást, azt a legutoljára létrehozott sorozatban egy speciális karakterrel, esetünkben a *-gal jelöljük
  - az alábbi táblázat mutatja az ábrával megadott gráf láncolt szomszédossági listáját



Elsőként létrehozzuk ezt a sorozatot ↘ Majd ezt ↘

2	1	1	3	2	1	4	4	5	3	4	4	és	1	2	6	3	11
4	12	5	7	10	8	*	*	*	9	*	*						

Végül ezt ↗

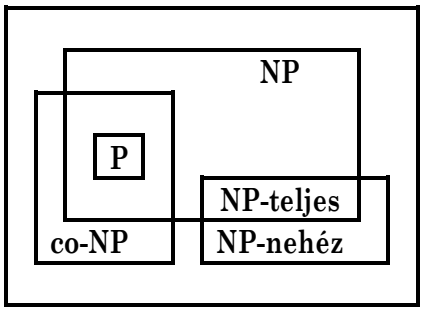
Gráftárolási módok összehasonlítása

	Sz. mátrix	Sz. tömb	Rendezett sz. lista	Láncolt sz. lista	Illeszkedési mátrix
Tárigény	v^2	$2e+v$	$2e+v$	$4e+v$	ev
2 P sz. ?	1	Δ	$\log\Delta$	Δ	1
P sz. megj.	v	Δ	Δ	Δ	v
$\forall$ él megj.	v^2	e	e	e	e
él hozzáadása	1	e	e	1	v
él elvétele	1	e	e	Δ	v
P elvétele	v	e	e	$\min(e;\Delta^2)$	e

Jelölések: v – pontszám, e – élszám, Δ – maximális fokszám, *sz.* – szomszéd... (különböző toldalékokkal), *megj.* – megjelölése, P – pont

14. Problémák bonyolultsága, polinomiális visszavezetés, P, NP, co-NP bonyolultsági osztályok fogalma, feltételezett viszonyuk, NP-teljesség, nevezetes NP-teljes problémák

- eldöntési problémának nevezzük az olyan problémákat, amikor az input egy eldöntendő kérdés
- az eldöntési problémák azon osztályát, amelyek az input méretének polinomiális függvényében felülről becsülhető időben megoldhatók, jelöljük **P**-vel
- ez gyakorlatilag azt jelenti, hogy a probléma megoldható véges idő alatt
- sajnos vannak olyan problémák, amelyekre eddig senki sem tudott polinomiális algoritmust adni
- ilyen például az, hoogy van-e a *G* gráfban Hamilton–kör
- ha valaki mutat nekünk egy Hamilton–kört, akkor arról polinom időben megállapítható, hogy tényleg az-e
- ha azonban nincs a gráfban Hamilton–kör, akkor ennek bizonyítására minden egyes körről meg kell mutatni, hogy nem Hamilton–kör, ezt viszont nyilvánvalóan nem végezhető el polinom időben
- az olyan eldöntési problémák osztályát, amelyeknél az igenlő választ polinom időben be tudjuk bizonyítani, **NP**-nek, amelyeknél a nemlegest, **co-NP**-nek nevezzük
- az eddigiekből nyilvánvaló, hogy $\mathbf{P} \subseteq \mathbf{NP}$
- a bonyolultságelmélet talán legérdekesebb megoldatlan kérdése, hogy itt valódi tartalmazás vagy egyenlőség áll fenn



- az ábra a kérdéskörben szereplő problémaosztályok legvalószínűbb viszonyát mutatja, látható, hogy vannak problémák, amelyek $\mathbf{NP} \cap \mathbf{co}-\mathbf{NP}$-ben vannak, ilyenek:
  (1) Van-e a *G* páros gráfban teljes párosítás?
  (2) Egy hálózatban van-e legalább *k* értékű folyam?
  (3) A *G* gráf síkbarajzolható-e?
- ezek valóban **NP**-ben vannak, hiszen ha a varázsló megmondja nekünk a párosítást, a folyamot, illetve a síkba lerajzolt gráfot, akkor ezek könnyen ellenőrizhetők
- ezek a problémák egyébként nemcsak $\mathbf{NP} \cap \mathbf{co}-\mathbf{NP}$-ben, hanem **P**-ben is benne vannak
- vannak, akik azt sejtik, hogy $\mathbf{NP} \cap \mathbf{co}-\mathbf{NP} = \mathbf{P}$, hiszen még egyetlen egy ilyen problémáról sem látták be, hogy nincs **P**-ben, és legtöbb $\mathbf{NP} \cap \mathbf{co}-\mathbf{NP}$-beli problémáról előbb–utóbb kiderült, hogy **P**-ben van
- tegyük fel, hogy most egy olyan számítógépünk van, amely a szokásos műveletek mellett még egy P_2 problémát is meg tud oldani egységnyi idő alatt
- ha evvel a számítógéppel polinom időben meg tudjuk oldani a P_1 problémát, akkor azt mondjuk, hogy P_1 polinomiálisan visszavezethető a P_2 problémára
- ha $P_2 \in \mathbf{P}$, akkor nyilván $P_1 \in \mathbf{P}$

- egy problémát **NP–nehéz**nek nevezünk, ha minden **NP**-beli probléma visszavezethető rá
- ha ez a probléma maga is eleme **NP**-nek, akkor **NP-teljes**nek hívjuk
- ha egy **NP**-teljes problémát meg tudnánk oldani polinom időben, akkor minden **NP**-beli probléma is megoldható lenne polinom időben
- Cook bebizonyította, hogy létezik **NP–teljes** probléma (pl.: Hamilton–kör)
- ha van egy problémánk, amire nem találunk megoldást, polinomrendű algoritmust, viszont nyilván **NP**-ben van, akkor megpróbálunk visszavezetni a mi problémánkra egy **NP–teljes** problémát
- ez azt jelenti, hogy a mi problémánk is **NP–teljes**, így a problémánk nem megoldható

Nevezetes polinomrendű illetve NP–teljes feladatok:

P–beli	**NP–teljes**
Van-e *G*-ben minimum *k* darab független él?	Van-e *G*-ben minimum *k* darab független pont?
Lefogható-e *G* minden pontja legfeljebb *k* éllel?	Lefogható-e *G* minden éle legfeljebb *k* ponttal?
Van-e *G*-ben maximum *k* hosszúságú út?	Van-e *G*-ben minimum *k* hosszúságú út?
Van-e *G*-ben maximum *k* hosszúságú kör?	Van-e *G*-ben minimum *k* hosszúságú kör?
Kiszínezhetők-e *G* pontjai legfeljebb 2 színnel?	Kiszínezhetők-e *G* pontjai legfeljebb 3 színnel?
Van-e egy hálózatban legfeljebb *k* értékű vágás?	Van-e egy hálózatban legalább *k* értékű vágás?
Van-e egy egytermékes hálózatban legalább egy *k* értékű folyam?	Van-e egy többtermékes hálózatban legalább *k* értékű folyam?

15. Oszthatóság, legnagyobb közös osztó, legkisebb közös többszörös, prímek, felbonthatatlanok, a számelmélet alaptétele, osztók száma, euklideszi algoritmus, nevezetes tételek prímszámokról

- legyenek $a,b \in \mathbb{Z}$; azt mondjuk, hogy *b* **osztható** *a*-val vagy *a* osztója *b*-nek (*a* osztja *b*-t), ha van olyan $q \in \mathbb{Z}$, amelyre $b = aq$; jelölése $a \mid b$
- **prímszám**nak nevezzük azokat az egynél nagyobb számokat, melyeknek nincs valódi osztóik
- oszthatóság szempontjából a negatív számok ugyanúgy viselkednek mint a pozitívak, *a* és *–a* osztói megegyeznek, valamint ha $a \mid b$, akkor $-a \mid b$ is igaz
- ezért ezen túl csak a nem negatív számokkal foglalkozunk, számon mindig nem negatív egész (természetes) számot értünk
- a p_0-tól és 1-től különböző egész számot **felbonthatatlan**nak nevezzük, ha $a,b \in N$, $p = ab$ esetén $p = b$
- a természetes számok körében a felbonthatatlanok megegyeznek a prímekkel
- **a számelmélet alaptétele** kimondja: minden egynél nagyobb szám a sorrendtől eltekintve egyértelműen előáll prímszámok szorzataként, azaz tetszőleges $n \in N$ esetén
  $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k} = \prod_{i=1}^{k} \left(p_i^{\alpha_i} \right)$ ahol a p_i-k prímszámok és az $\alpha_i > 0$ kitevők egyértelműek
- a fenti fölbontást *n* kanonikus alakjának nevezzük
- az $\alpha_i > 0$ kikötésre azért van szükség, mert különben tetszőleges *n*-et nem osztó prímeket hozzávehetnénk a felbontáshoz 0 kitevővel, ekkor a felbontás nem lenne egyértelmű
- ennek alapján könnyen meghatározhatók a szám osztói: legyen $n = \prod_{i=1}^{k} \left(p_i^{\alpha_i} \right)$ és $d \mid n$, ekkor *d* előáll $d = \prod_{i=1}^{k} \left(p_i^{\beta_i} \right)$ alakban, ahol $\beta_i \le \alpha_i$
- minden ilyen alakú szám osztója *n*-nek
- a számok kanonikus alakjának a segítségével meghatározható két szám legnagyobb közös osztója és legkisebb közös többszöröse
- legyenek $n,m \in \mathbb{N}$; *n* kanonikus alakját „pótoljuk” ki az *m*-ben szereplő, *n*-ben nem szereplő prímekkel 0 kitevővel és fordítva
- ekkor mindkét szám fölírásában ugyanazok a prímek szerepelnek
- legyen $n = \prod_{i=1}^{k} \left(p_i^{\alpha_i} \right)$ és $m = \prod_{i=1}^{k} \left(p_i^{\beta_i} \right)$, ekkor *n* és *m* legnagyobb közös osztója:
  $(n,m) = \prod_{i=1}^{k} \left(p_i^{\min\{\alpha_i,\beta_i\}} \right)$, valamint legkisebb közös többszöröse: $[n,m] = \prod_{i=1}^{k} \left(p_i^{\max\{\alpha_i,\beta_i\}} \right)$
- az *a*, *b* számokat **relatív prím**eknek nevezzük, ha legnagyobb közös osztójuk 1
- az *n* szám osztóinak számát $d(n)$-nel, osztóinak összegét $\sigma(n)$-nel, a nála kisebb, hozzá relatív prím számok számát $\varphi(n)$-nel jelöljük
- legyen $n = \prod_{i=1}^{k} \left(p_i^{\alpha_i} \right)$, ekkor:

$$d(n)=\prod_{i=1}^{k}(\alpha_i+1) \qquad \sigma(n)=\prod_{i=1}^{k}\left(\frac{p_i^{\alpha_i+1}-1}{p_i-1}\right) \qquad \varphi(n)=n\prod_{i=1}^{k}\left(1-\frac{1}{p_i}\right)$$

- mindhárom függvény rendelkezik az úgynevezett **multiplikatív tulajdonság**gal
- az *f* egészeken értelmezett függvény multiplikatív, ha $(a,b)=1$ esetén $f(ab)=f(a)\cdot f(b)$
- a *d, σ, φ* függvények multiplikativitása a rájuk vonatkozó képletekből látszik
- **euklideszi algoritmus**: polinomrendű, az adott két *a, b* egész szám legnagyobb közös osztóját lehet meghatározni vele
- ha $a>b$ akkor az $a:b$ maradékos osztást elvégezzük, majd *b*-t osztjuk a maradékkal stb.:

$$\begin{aligned} a &= h_1 b + m_1 && (0 \le m_1 < b) \\ b &= h_2 m_1 + m_2 && (0 \le m_2 < m_1) \\ m_1 &= h_3 m_2 + m_3 && (0 \le m_3 < m_2) \\ &\cdot && \cdot \\ &\cdot && \cdot \\ &\cdot && \cdot \end{aligned}$$

- a *k*. lépésben a hányadost h_k-val, a maradékot m_k-val jelöljük
- az eljárás akkor ér véget, ha nincs az osztásnak maradéka, vagyis: $m_{n-2}=h_n m_{n-1}$
- ekkor persze $m_{n-3}=h_{n-1}m_{n-2}+m_{n-1}=(h_{n-1}h_n+1)m_{n-1}$, és ugyanígy visszahelyettesítve *a* is, *b* is m_{n-1} konstans-szorosa lesz
- belátható, hogy épp $m_{n-1}=d(a,b)$

16. Kongruencia fogalma, teljes és redukált maradékrendszer, φ-függvény, Euler–Fermat–tétel, kis–Fermat–tétel, lineáris kongruenciák megoldása, Wilson–tétel

A kongruencia fogalma

- legyen $m > 1$ egy rögzített egész szám; akkor azt mondjuk, hogy *a* kongruens *b*-vel az *m* modulusra vonatkozólag (jelölés: $a \equiv b \pmod{m}$ vagy $a \equiv b \ (n)$), ha az *a* és *b* számok *m*-mel osztva ugyanazt a maradékot adják
- a kongruencia ekvivalenciareláció
- az állítás közvetlenül látszik, ha észrevesszük, hogy $a \equiv b \pmod{n}$ pontosan akkor, ha $n \mid a-b$
- legyenek most *a, b, c* egész számok, az $ax \equiv b \pmod{c}$ kongruencia megoldhatóságát fogjuk vizsgálni, ez azt jelenti, hogy keressük azon *x* egész számokat, amelyekre $c \mid ax-b$
- nyilván, ha *x* kielégíti a feltételt, akkor $kc+x$ is tetszőleges *k* egész számra, így a megoldásokat modulo *c*-re fogjuk keresni
- a feltétel azzal ekvivalens, hogy van olyan $y \in \mathbb{Z}$, amelyre $cy = ax-b$, azaz $cy-ax=b$
- a kétféle átfogalmazásból közvetlenül igazolható, hogy az $ax \equiv b \pmod{c}$ kongruenciában *a* és *b* osztható és megszorozható tetszőleges *c*-hez tartozó relatív prím számmal, *a, b, c* pedig osztható tetszőleges egész számmal, a kongruencia megoldáshalmaza nem változik
- a kongruencia pontosan akkor oldható meg, ha $(a,c) \mid b$, a megoldások száma $(a,c) \pmod{c}$

Maradékrendszerek

- a kongruencia ekvivalenciareláció, amely osztályokba sorolja az egész számok halmazát
- egy-egy ilyen osztályt hívunk **maradékosztály**nak, más szóval egy osztályt alkot az összes *m*-mel osztható szám, egy osztályt alkotnak az *m*-mel osztva egy maradékot adók, stb.
- ha a mod *m* maradékosztályok mindegyikéből kiválasztunk egy tetszőleges elemet, a keletkező számhalmazt mod *m* **teljes maradékrendszer**ének nevezzük
- egy $\{b_1, b_2, \ldots, b_n\}$ számhalmaz akkor és csak akkor alkot mod *m* teljes maradékrendszert, ha
  (1) $n = m$
  (2) bármely $i \neq j$ indexpárra $b_i \not\equiv b_j \pmod{m}$
- ha két szám ugyanabba a mod *m* maradékosztályba tartozik, akkor vagy mindkettő relatív prím *m*-hez, vagy egyik sem
- ez alapján a szempont alapján két csoportba oszthatjuk a mod *m* maradékosztályokat
  (1) azokra, amelyek minden eleme relatív prím *m*-hez, és
  (2) azokra, melyeknek egyik eleme sem
- az (1)-es csoportba épp annyi mod *m* maradékosztály tartozik, ahány szám a $\{0,1,2,\ldots,m-1\}$ halmazból relatív prím *m*-hez, ezt a számot $\varphi(m)$-el jelöljük
- ha az első csoportba tartozó minden maradékosztályból kiválasztunk egy tetszőleges elemet, a keletkező számhalmazt mod *m* **redukált maradékrendszer**nek nevezzük
- egy $\{c_1, c_2, \ldots, c_n\}$ számhalmaz akkor és csak akkor alkot mod *m* redukált maradékrendszert, ha
  (1) $k = \varphi(m)$

(2) bármely $i \neq j$ indexpárra $c_i \not\equiv c_j \pmod{m}$

(3) bármely *i* indexre $d(c_i, m) = 1$

– legyen $d(a,m) = 1$, ha egy mod *m* teljes–, vagy redukált maradékrendszer minden elemét *a*-val megszorzunk, ismét egy mod *m* teljes–, vagy redukált maradékrendszert kapunk

φ-függvény

– az n számhoz relatív prím számok számát $\varphi(n)$-nel jelöljük, $\varphi(n) = n\prod_{i=1}^{k}\left(1 - \frac{1}{p_i}\right)$, ahol p_i-k a prímtényezők, ha *p* prím, $\varphi(p) = p - 1$

Euler–Fermat–tétel

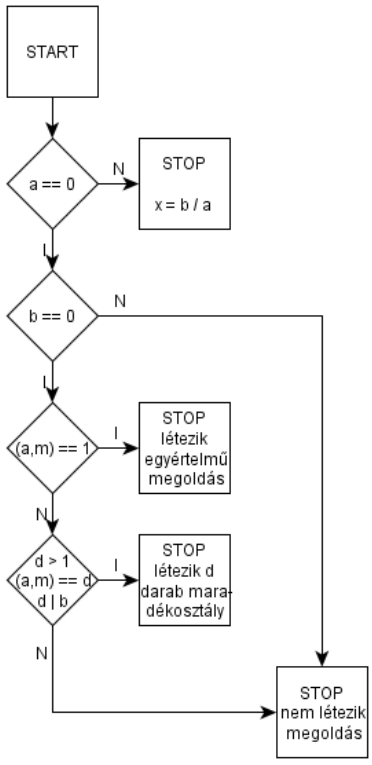
– ha $m > 1$ tetszőleges egész szám és *a* tetszőleges olyan szám, melyre $d(a,m) = 1$, akkor $a^{\varphi(m)} \equiv 1 \pmod{m}$

kis–Fermat–tétel

– tetszőleges *p* prímszámra és tetszőleges *a* egész számra $a^p \equiv a \pmod{p}$

Lineáris kongruenciák megoldása

– adott a következő lineáris kongruencia: $ax \equiv b \pmod{m}$, ekkor két lehetőség van:
  – $a \neq 0$: egyértelmű megoldás: $x = \frac{b}{a}$
  – $a = 0$:
    – $b \neq 0$: nem létezik megoldás
    – $b = 0$: több megoldás van
      – ha $(a,m) = 1$, $ax \equiv b \pmod{m}$ egyértelműen megoldható
      – ha $(a,m) = d > 1$:
        – ha *d* nem osztója *b*-nek, akkor nem létezik megoldás
        – ha $d \mid b$, akkor *d* db maradékosztály létezik (mod *m*)



Példák lineáris kongruenciák megoldására

(1)

$3x \equiv 1 \pmod{5}$

$(3,5) = 1$

$\varphi(5) = 4 \Rightarrow 3^4 \equiv 1 \pmod{5}$

$3 \cdot 3^3 x \equiv 3^3 \pmod{5} \Rightarrow x \equiv 3^3 \pmod{5} \equiv 2 \pmod{5}$

(2)

$3x \equiv 12 \pmod 6$

$(3,6) = 3$, de 3 nem osztója 1-nek, ezért ennek nincs megoldása

(3)

$3x \equiv 12 \pmod{15}$

$\left.\begin{array}{l}(3,15) = 3 \\ 3 \mid 12\end{array}\right\} \Rightarrow \exists$ 3 db megoldás

Osszunk hárommal!

$$x \equiv 4 \left(\bmod \frac{15}{(15,3)} \right)$$

$x \equiv 4 \pmod 5$

Megoldások:

$$\left.\begin{array}{l}x_1 = 4 \\ x_2 = 9 \\ x_3 = 14\end{array}\right\} \pmod{15}$$

Wilson–tétel

– legyen $k \geq 2$ tetszőleges szám, ekkor $(k-1)! \equiv \begin{cases} -1 \pmod k, \text{ ha } k \text{ prím} \\ 2 \pmod k, \text{ ha } k = 4 \\ 0 \pmod k, \text{ ha } k \geq 6 \text{ összetett szám} \end{cases}$

17. Félcsoportok, csoportok, példák, csoport rendje, elem rendje, szimmetrikus idomok egybevágósági transzformációinak csoportja, ciklikus csoport, az S_n szimmetrikus csoport

- legyen H tetszőleges halmaz, jelölje H^n a H halmaz elemeiből képzett n hosszú sorozatokat
- az $f : H^n \to H$ mindenütt értelmezett függvényt n változós **művelet**nek nevezzük
- kétváltozós művelet pl. az egész számok összeadása ($f(a,b) = a + b$)
- ilyenkor az $f(a,b)$ jelölés helyett a két elem közé beírjuk a műveleti jelet
- háromváltozós pl. a vektorok vegyes szorzata
- nem művelet a pozitív számok kivonása, hiszen, ha $a \le b$, akkor nincs értelmezve
- ($f(2,3) = 2 - 3$ nem pozitív, tehát nincs értelme az adott halmazon)
- egy H halmazon értelmezett kétváltozós műveletet (jelöljük $*$-al) **kommutatív**nak nevezünk, ha $(\forall a,b \in H)(a * b = b * a)$ és **asszociatív**nak nevezünk, ha
  $(\forall a,b,c \in H)\left[(a * b) * c = a * (b * c)\right]$
- az S halmazt a rajta értelmezett $*$ művelettel **félcsoport**nak nevezzük, ha $*$ asszociatív
- ha $*$ kommutatív is, akkor **kommutatív (vagy Abel–féle) félcsoport**ról beszélünk
- példák:
  - a pozitív számok az összeadásra nézve félcsoportot alkotnak
  - a pozitív valós számok a szorzásra nézve félcsoportot alkotnak
- ha $(\exists e \in S)(\forall a \in S)(e * a = a * e = a)$, akkor e-t **neutrális–, vagy egységelem**nek hívjuk, S-et pedig **egységelemes félcsoport**nak nevezzük
- egy G halmazt a $\cdot$ művelettel **csoport**nak nevezzük, ha
  - $(\forall a,b,c \in G)\left[(a \cdot b) \cdot c = a \cdot (b \cdot c)\right]$ (a művelet asszociatív)
  - $(\exists e \in G)(\forall a \in G)(e * a = a * e = a)$ (létezik egységelem)
  - $(\exists a^{-1} \in G)(\forall a \in G)(a \cdot a^{-1} = a^{-1} \cdot a = e)$ (létezik a^{-1}, az a elem inverze)
- ha a csoportban teljesül a kommutativitás, akkor **kommutatív (vagy Abel–féle) csoport**ról beszélünk
- a csoport elemszámát $|G|$-vel jelöljük, és G **rend**jének nevezzük
- példák:
  - az egész, a racionális és a valós számok Abel–csoportot alkotnak az összeadásra nézve: $(\mathbb{Z},+)$, $(\mathbb{Q},+)$, $(\mathbb{R},+)$, a természetes számok $(\mathbb{N})$ viszont nem
  - a pozitív valós és pozitív racionális számok Abel–csoportot alkotnak a szorzásra nézve: $(\mathbb{R}^+,\cdot)$, $(\mathbb{Q}^+,\cdot)$
  - a szabályos n-szög egybevágóságai csoportot alkotnak, ahol a művelet az egymás után való elvégzés (a csoport egységeleme a helybenhagyás, a csoport rendje $2n$, ugyanis van n darab tengelyes tükrözés és a helybenhagyással együtt n forgatás, a csoportot D_n-nel jelöljük és **diédercsoport**nak hívjuk)
  - n elem permutációi (önmagára való bijektív leképzései) csoportot alkotnak a kompozícióra, a csoportot n-ed fokú **szimmetrikus csoport**nak nevezzük, S_n-nel jelöljük, rendje $n!$; egy H halmaz elemeinek összes permutációinak csoportját S_H-val jelöljük

- a szimmetrikus csoport esetében a következőképpen értelmezzük a műveletet: legyen általánosságban $\begin{pmatrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n \end{pmatrix}$ a permutáció jelölése, ahol 1 képe i_1, 2 képe i_2, stb.
- tekintsük konkrét példaként S_3-at, melynek elemeit jelöljük így:
  (1) ABC
  (2) ACB
  (3) BAC
  (4) BCA
  (5) CAB
  (6) CBA
- ekkor pl. a (2)-es és (4)-es elemek közt végzett műveletet a következőképpen értelmezzük: $\begin{pmatrix} A & B & C \\ A & C & B \end{pmatrix} \cdot \begin{pmatrix} A & B & C \\ B & C & A \end{pmatrix} = \begin{pmatrix} A & B & C \\ B & A & C \end{pmatrix}$, ugyanis azt nézzük, hogy a permutációkkal milyen új elemhez jutunk, tehát konrétan:
  - az első tényezőben *A* képe *A*, tehát a második tényezőben *A* képét nézzük, ami *B*, vagyis *A* önmagán keresztül *B*-be „vitte” a permutációt, így az eredményben *A* képe *B*
  - a második tényezőben *B* képe *C*, tehát a második tényezőben *C* képét nézzük, ami *A*, vagyis *B C*-n keresztül *A*-ba „vitte” a permutációt, így az eredményben *B* képe *A*
  - az első tényezőben *C* képe *B*, tehát a második tényezőben *B* képét nézzük, ami *C*, vagyis *C B*-n keresztül önmagába „vitte” a permutációt, így az eredményben *C* képe *C*

- következmények:
  - az egységelem egyértelmű: tegyük fel, hogy *e'* és *e"* különböző egységelemek, ekkor $e' = e' \cdot e'' = e''$
  - az inverz egyértelmű: legyenek *a'* és *a" a* két különböző inverze, ekkor
    $a' \cdot a'' = (a'' \cdot a) \cdot a' = e \cdot a' = a'$
    $a'' \cdot a \cdot a' = a'' \cdot (a \cdot a') = a' \cdot e = a''$
    $\Rightarrow a' = a''$

- átfogalmazások:
  - a megkövetelt egység–, és inverz elem helyett elég egyoldali egység–, illetve inverz elem létezését előírni:
    - egy G halmaz a · művelettel pontosan akkor csoport, ha
      - $(\forall a,b,c \in G)\left[(a \cdot b) \cdot c = a \cdot (b \cdot c)\right]$
      - $(\exists e \in G)(\forall a \in G)(e * a = a * e = a)$
      - $(\exists e_0 \in G)(\forall a \in G)(\exists a^{-1} \in G)(a \cdot a^{-1} = e_0)$
    - legyen *a' a* egyik jobbinverze,
    - $a \cdot a' = e_0 = e_0 \cdot e_0 = e_0 \cdot a \cdot a'$
    - az $a \cdot a' = e_0 \cdot a \cdot a'$ egyenlőség mindkét oldalát megszorozva *a'* jobboldali inverzével az $a = e_0 \cdot a$ egyenlőséget kapjuk, ezért e_0 baloldali egységelem is
    - tehát e_0 (kétoldali) egységelem, és mint ilyen, a fentiek miatt egyértelmű
    - $a' = a' \cdot e_0 = a' \cdot a \cdot a'$

  - mindkét oldalt jobbról szorozva *a'* valamely jobboldali inverzével az $e_0 = a \cdot a'$ összefüggést kaptuk, tehát *a'* balinverz is
- ezen átfogalmazás előnye, hogy nem szerepelnek benne kitüntetett elemek
  - egy asszociatív művelettel ellátott *G* struktúra pontosan akkor csoport, ha tetszőleges *a, b* *G*-beli elemekhez található *G*-ben egyetlen olyan *x* és egyetlen olyan *y* elem, amelyekre $b = y \cdot a = y \cdot (a \cdot e_a) = (y \cdot a) \cdot e_a = b \cdot e_a$, azaz
    - $e_a = e$ a csoport jobboldali egységeleme
    - az $a \cdot x = e$ egyenlet megoldása a jobbinverze
    - ezzel igazoltuk az előző állítás feltételeit, tehát *G* csoport
- a G_1, G_2 csoportokat **izomorf**aknak nevezzük, ha van köztük egy kölcsönösen egyértelmű művelettartó leképzés, azaz van olyan $\phi : G_1 \to G_2$ leképzés, amely bijektív és $\forall g,h \in G_1$ esetén a következő teljesül: $\phi(g)\phi(h) = \phi(gh)$, jelölése $G_1 \cong_\phi G_2$, vagy egyszerűen $G_1 \cong G_2$
- a részcsoportnak példái az **egy elem által generált részcsoport**ok, vagyis a **ciklikus csoprt**ok
  - legyen $a \in G$, ekkor $\langle a \rangle$ nyilván tartalmazza *aa*-t, *aaa*-t, stb.
  - az *a* elem *n*-szer önmagával vett szorzatát a^n-nel jelöljük
  - ekkor természetesen igaz, hogy $a^{n+k} = a^n a^k$ és $(a^n)^k = a^{nk} \quad \forall n,k \in \mathbb{Z}^+$
  - $\langle a \rangle$ továbbá tartalmazza a^{-1}-et is, valamint ennek hatványait
  - tekintsük az $(a^{-1})^n a^n$ szorzatot: kiírva tényezőnként azt kapjuk, hogy a szorzat értéke *e*, a csoport egységeleme, tehát $(a^{-1})^n = (a^n)^{-1}$, jelöljük ezt az elemet a^{-n}-nel
  - a hatványozás tulajdonságai ezek alapján kiterjeszthetőek negatív hatványokra is: $a^{n+k} = a^n a^k$ és $(a^n)^k = a^{nk} \quad \forall n,k \in \mathbb{Z}$
  - ezek szerint $\langle a \rangle = \{a^n \mid n \in \mathbb{Z}\}$, azaz az egy elem által generált részcsoport az elem (negatív és pozitív kitevős) hatványaiból áll
  - különböztessünk meg két esetet:
    - a összes hatványa különböző
    - $(\exists k,l \in \mathbb{Z})(a^k = a^l) \Rightarrow a^{k-l} = 1$, azaz van *a*-nak olyan hatványa, amely az egységelem; legyen *n* a legkisebb ilyen szám
  - a legkisebb ilyen számot az **elem rendjének** nevezzük, és $o(a)$-val jelöljük („ordó *a*"); ha nincs ilyen szám, végtelen rendű elemről beszélünk
  - $o(a) = n \Rightarrow \langle a \rangle = \{1, a, a^2, ..., a^{n-1}\}$
  - ezen elemek különbözőek, mert $(a^j = a^i)(i > j)$ esetén $a^{i-j} = 1$ lenne, ahol $i - j < n$
  - továbbá $\forall k \in \mathbb{Z}$ előáll $k = qn + r$ alakban, ahol $0 \le r < n$, és $a^k = a^{qn+r} = a^{qn} a^r = (a^n)^q a^r = 1^q a^r = a^r$, tehát *a* minden hatványa szerepel az első $n-1$ között
  - ezzel igazoltuk, hogy jogos volt két látszólag távoli fogalomra ugyanazt a szót használni:
  - egy elem rendje megegyezik az általa generált részcsoport rendjével
- következmény: minden $n > 0$ egész számra van *n* elemű csoport
- azonos rendű ciklikus csoportok izomorfak
- ciklikus csoport részcsoportja ciklikus

18. Részcsoport, mellékosztály, Lagrange tétele, elem és csoport rendjének kapcsolata, gyűrűk, nullosztó, példák, testek, példák

- legyen *G* egy csoport; egy $H \subseteq G$ részhalmazt **részcsoport**nak nevezünk, ha *H* is csoport ugyanarra a műveletre nézve; jelölése $H \leq G$
- minden csoportnak részcsoportja maga a csoport, és az egységelemet tartalmazó egyelemű halmaz; ezeket a részcsoportokat **triviális részcsoport**nak, az ezektől különböző részcsoportokat **valódi részcsoport**oknak nevezzük
- példák:
  - a valós számok additív csoportjának részcsoportja a racionális számok, annak pedig az egész számok additív csoportja
  - a háromszög egybevágóságának (D_3) részcsoportját alkotják a forgatások
  - *n* elem permutációinak részcsoportját alkotják a páros permutációk
  - a nem 0 komplex számok ($\mathbb{C}^*$) a szorzásra nézve csoportot alkotnak; ennek egy részcsoportját alkotják az 1 abszolút értékű komplex számok, annak pedig részcsoportját az *n*-edik egységgyökök
- annak ellenőrzése végett, hogy egy *G* csoport *H* részhalmaza részcsoport-e, elég leellenőrizni, hogy $a,b \in H$ esetén $a \cdot b$ és a^{-1} is *H*-beli-e, az asszociativitás ugyanis automatikusan teljesül a csoportaxiómák miatt, az egységelemet pedig megkapjuk, ha valamely *H*-beli elemet megszorozzuk az inverzével
- részcsoportok metszete is részcsoport, azaz legyenek $H_i \leq G\left(i \in A\right)$, ahol *A* valamilyen indexhalmaz, akkor $\bigcap H_i$ is részcsoport
- legyen $K \subseteq G$; *K* által **generált részcsoport**nak nevezzük és $\langle K \rangle$-val jelöljük a *K*-t tartalmazó legszűkebb részcsoportot, ami nem más, mint a *K*-t tartalmazó részcsoportok metszete
- legyen $K, M \subseteq G$; a *KM* szorzaton a $KM = \{km \mid k \in K, m \in M\}$ halmazt értjük; legyen $H \leq G$ részcsoport, $g \in G$; a *Hg* (*gH*) szorzatot *H g* szerinti jobboldali (baloldali) **mellékosztály**ának, *g*-t pedig a mellékosztály **reprezentáns**ának nevezzük
- legyen $H \leq G$, ekkor:
  (1) $g \in Hg$
  (2) a *Hg* mellékosztály minden eleme reprezentálja a *Hg* mellékosztályt
  (3) két különböző jobboldali mellékosztály vagy egybeesik, vagy diszjunktak
  (4) ha *H* véges, akkor bármely mellékosztály elemszáma megegyezik *H* rendjével
- bizonyítás:
  (1) $1 \in H \Rightarrow g = 1g \in Hg$
  (2) legyen $h \in Hg$, ekkor $\left(\exists h_1 \in H\right)\left(h = h_1 g\right)$; a $\forall x \in H$-ra érvényes $xh = \left(xh_1\right)g$ összefüggés igazolja, hogy $Hh \subseteq Hg$ és $xg = xh_1^{-1}h$ pedig azt, hogy $Hg \subseteq Hh$
  (3) (1)-ből és (2)-ből közvetlenül következik
  (4) a $h_1 g = h_2 g$ egyenlőséget g^{-1}-el jobbról szorozva kapjuk, hogy $h_1 \neq h_2$ esetén $h_1 g$ és $h_2 g$ különbözőek
- bizonyítsuk **Lagrange tételét**, mely szerint, ha *G* véges, $H \leq G$, akkor *H* rendje osztja *G* rendjét
  - osztályozzuk *G*-t a *H* szerinti jobboldali mellékosztályok szerint: $G = \bigcup Hg$

  - jelölje *k* *H* mellékosztályainak számát
  - mivel minden elem pontosan egy mellékosztályban szerepel, ezért $|G| = \left|\bigcup Hg\right|$ miatt $|G| = \sum |Hg| = k|H|$
  - a $k = \frac{|G|}{|H|}$ számot *H* *G*-beli indexének nevezzük és $|G:H|$-val jelöljük, $|G:H||H| = |G|$
  - mivel egy elem rendje megegyezik az általa generált részcsoport rendjével: egy elem rendje osztja a csoport rendjét
- ha $|G| = p$, *p* prím, ekkor egy egységelemtől különböző csoportelem által generált ciklikus csoport rendje csak *p* lehet, azaz minden prímrendű csoport ciklikus
- legyen *G* csoport, $N \le G$; *N* **normálosztó** *G*-ben ($N \lhd G$), ha *N* jobboldali és baloldali mellékosztályai megegyeznek
- ez azt jelenti, hogy minden *Nh* mellékosztály előáll h_1N alakban
- mivel $h \in Nh$ és $h \in hN$ ez csak úgy lehet, ha $(hN = Nh)(\forall n \in G)$
- az alábbi állítások ekvivalensek:
  (1) $N \lhd G$
  (2) $(gN = Ng)(\forall g \in G)$
  (3) $(g^{-1}Ng = N)(\forall g \in G)$
  (4) $(\forall h \in N)(\forall g \in G)(g^{-1}hg \in N)$
- bizonyítás
  - $(1) \leftrightarrow (2)$: világos
  - $(2) \leftrightarrow (3)$: szorozzuk meg $gN = Ng$ egyenlőség mindkét oldalát balról g^{-1}-el
  - $(3) \rightarrow (4)$: nyilvánvaló
  - $(4) \rightarrow (3)$: $g^{-1}Ng \subseteq N$, valamint $gNg^{-1} \subseteq N$, ez utóbbi tartalmazási relációt jobbról *g*-vel, balról g^{-1}-el szorozva $N \subseteq g^{-1}Ng$-t kapjuk, amiből következik (3)

Gyűrűk, testek

- az *R* halmaz a + és · műveletekkel **gyűrű**, ha
  (1) $(a+b=b+a)(\forall a,b \in R)$
  (2) $\left[(a+b)+c=a+(b+c)\right](\forall a,b,c \in R)$
  (3) $(\exists 0 \in R)(a+0=0+a=a)(\forall a \in R)$
  (4) $(\forall a \in G)(\exists a' \in G)(a+a'=0)$
  (5) $\left[(a \cdot b) \cdot c = a \cdot (b \cdot c)\right](\forall a,b,c \in R)$
  (6) $\left[(a+b) \cdot c = a \cdot c + b \cdot c\right](\forall a,b,c \in R)$
  (7) $\left[c \cdot (a+b) = a \cdot c + b \cdot c\right](\forall a,b,c \in R)$
- az első négy axióma azt mondja ki, hogy *R* Abel–csoport az összeadásra nézve, az (5) pedig, hogy félcsoport a szorzásra nézve
- a szorzás · -ját gyakran elhagyjuk
- a (6), illetve (7) axiómákat jobboldali illetve baloldali **disztributív törvény**nek nevezzük

- ha a szorzás is kommutatív, **kommutatív gyűrű**ről, ha van a szorzásra nézve egységelem, **egységelemes gyűrű**ről beszélünk
- a harmadik axiómában említett elemet **nullelem**nek nevezzük
- egy a *R*-beli elem összeadásra vonatkozó inverzét ((4) axióma) **ellentettnek** hívjuk, és *–a*-val jelöljük
- az $a-b=a+(-b)$ műveletet **kivonás**nak nevezzük
- az axiómák közvetlen következményei arról szólnak, hogy egy gyűrűben teljesülnek azok a műveleti tulajdonságok, amiket elvárunk egy gyűrűtől:
- legyen *R* gyűrű, $a,b\in R$:
  (1) a nullelem és az ellentett egyértelmű
  (2) $0a=a0=0$
  (3) $(-a)b=-ab$
  (4) $(-a)(-b)=ab$
- bizonyítás
  (1) *R* Abel–csoport az összeadásra
  (2) az $a0=a(0+0)=a0+a0$ egyenlőség mindkét oldalához *a0* inverzét adva
  (3) $ab+(-a)b=(a-a)b=0b=0$ egyenlőség következménye
  (4) $(-a)(-b)=-\left[a(-b)\right]=-(-ab)=ab$
- egy *R* egységelemes gyűrűt **ferdetest**nek hívunk, ha a szorzásra nézve is balinverz, azaz $(\forall 0\neq a\in R)(\exists a'\in R)(aa'=1)$
- egy ferdetestet **test**nek nevezünk, ha a szorzás kommutatív
- legyen *R* gyűrű; a $(\forall 0\neq a\in R)$ elemet baloldali (jobboldali) **nullosztó**nak nevezzük, ha $(\exists 0\neq b\in R)(ab=0)$ (vagy jobboldali esetén: $ba=0$)
- a kommutatív, nullosztómentes gyűrűt **integritási tartomány**nak nevezzük
- minden ferdetest nullosztómentes

19. Számelméleti algoritmusok, prímtesztelés, nyilvános kulcsú titkosítás, bizonyítás információközlés nélkül

Számelméleti algoritmusok

	egész számok	(mod *m*) maradékosztályok
+	lineáris	polinom
–	lineáris	polinom
·	polinom	polinom
:	polinom	polinom
hatványozás	exponenciális	polinom

input: *a, b*, kérdés: $a+b=?$, input hossza: $\log a$

input: *a*, kérdés: 2^a, output hossza: $\log\left(2^a\right)=ca=c2^n$, input hossza: $\log a=n$

$2^{100}\equiv ?\pmod 5$

$2^2\equiv 4\pmod 5$

$2^4\equiv 1\pmod 5$

$2^8\equiv 1\pmod 5$

$2^{16}\equiv 1\pmod 5$

$2^{32}\equiv 1\pmod 5$

$2^{64}\equiv 1\pmod 5$

$2^{100}=2^{64+32+4}=2^{64}\cdot 2^{32}\cdot 2\equiv 1\pmod 5$

$(100)_{10}=(1100100)_2$

$a^b\equiv ?\pmod c\Rightarrow a^2,\ a^4,\ a^8,\ a^{16},\ldots\Rightarrow \log b$ darab hatványozás + $\log b$ darab szorzás

input: *a, b*, kérdés: $d(a,b)=?$ (az Euklideszi algoritmus polinom rendű)

$a=h_1b+m_1$

$b=h_2m_1+m_2$

$m_1=h_3m_2+m_3$

.

.

.

$m_n=h_{n+2}m_{n+1}$

Például:

$13=1\cdot 8+5$

$8=1\cdot 5+3$

$5=1\cdot 3+2$

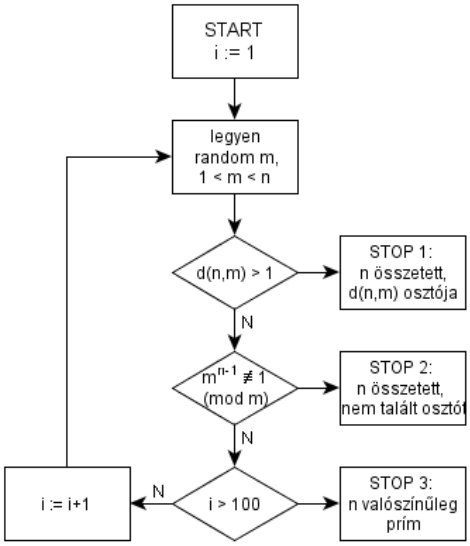
$3=1\cdot 2+1$

$2=2\cdot 1+0$

Prímtesztelés

- **Eratoszthenész „szita–algoritmusa”**:
  - el lehet dönteni vele, hogy egy szám prím-e
  - írjuk fel az egész számokat 2-től *n*-ig, húzzuk ki (vagyis szitáljuk ki) a páros számokat, kivéve a 2-t, azután a maradékból a hárommal oszthatókat, kivéve a 3-t, aztán az öttel oszthatókat, kivéve az 5-t stb.
  - minden ilyen lépés után a megmaradtak közül a legkisebb egy prím, őt hagyjuk meg, de a többszöröseit húzzuk ki
  - így elvileg bármilyen határig előbb-utóbb elő lehet állítani az összes prímet
  - egy másik lehetőség egyszerűen minden *n*-nél kisebb számról megnézni, nem osztója-e *n*-nek
  - a valóságban elég csak $\lfloor \sqrt{n} \rfloor$-ig próbálkozni, ha addig semmi nem osztója *n*-nek, akkor *n* biztos prímszám (hisz ha *k* osztója *n*-nek, akkor $\frac{n}{k}$ is osztója, és $\min\left(k;\frac{n}{k}\right) \le \lfloor \sqrt{n} \rfloor$)
  - az utóbbiról azonnal látszik, hogy lépésszám-igénye $\lfloor \sqrt{n} \rfloor$-nel arányos
  - figyelembe véve, hogy a feladat inputja *n*, az input hossza $\lceil \log n \rceil$, ennek a lépésszám nem polinomja
  - az eredeti Eratoszthenész–szitáról is belátható, hogy a lépésszám az input hosszának exponenciális függvénye
  - azért ezen algoritmusoknak van egy nagy előnyük is: ha ugyanis *n* nem prímnek, hanem összetett számnak bizonyul, akkor rögtön *n* valamely osztóját is megtalálják
- a következőkben egy olyan prímtesztelő algoritmust mutatunk be, mely polinom időben véget ér, de egyrészt az eredmény nem biztosan, hanem csak valószínűleg igaz, másrészt, ha összetett számnak tartja *n*-et, akkor általában nem találja meg egyetlen osztóját sem
  - az algoritmus alapgondolata az Euler–Fermat–tételt használja fel
  - ha *n* prím, tehát $\varphi(n) = n-1$, akkor $t^{n-1} \equiv 1 \pmod{n}$ teljesül minden olyan *t*-re, mely *n*-hez relatív prím
  - ha egy konkrét *t, n* párra $t^{n-1} \equiv 1 \pmod{n}$ nem teljesül, akkor minden további vizsgálat nélkül biztosak lehetünk abban, hogy *n* nem prím (tehát *t* az *n* „**áruló**ja”)
  - ha egyáltalán vannak egy összetett számnak árulói (általában vannak; nagyon kevés olyan, úgynevezett **Carmichael–szám** van, amelynek minden hozzá relatív prím szám **cinkos**a), akkor legalább annyi az árulója, mint a cinkosa (cinkos: segíti *n*-t abban, hogy megtévesszen minket és elhitesse velünk, hogy ő prím)
  - belátása: vegyük észre, hogy cinkosok szorzata cinkos, míg egy cinkos és egy áruló szorzata áruló
  - ha tehát $c_1, c_2, \ldots, c_s$ az összes cinkos sorozata, és *a* egy áruló, akkor az $ac_1, ac_2, \ldots, ac_s$ sorozat minden tagja áruló (és belátható, hogy mind különbözőek)
  - így már 2*s* darab különböző maradékosztályt találtunk, és lehet, hogy további árulók is vannak
  - ha tehát egymástól függetlenül véletlenszerűen választunk *q* darab maradékosztályt, és mindegyikre $m^{n-1} \equiv 1 \pmod{n}$ teljesül, akkor *n* lehet ugyan összetett szám, de ennek a valószínűsége $\left(\frac{1}{2}\right)^q$-nál kisebb (vagyis elég nagy *q* érték mellett szinte lehetetlen)

- így az alábbi algoritmus – néhány pszeudoprímtől eltekintve – minden *n* inputra működik:
  - 1. lépés: $i := 1$
  - 2. lépés: válasszunk véletlenszerűen egy $1 < m < n$ számot
  - 3. lépés: határozzuk meg *m* és *n* legnagyobb közös osztóját, ha ez $> 1 \rightarrow$ STOP 1
  - 4. lépés: ha $m^{n-1} \not\equiv 1 \pmod{n} \rightarrow$ STOP 2
  - 5. lépés: ha $i > 100 \rightarrow$ STOP 3
  - 6. lépés: $i := i + 1$ és folytassuk az 2. lépésnél
  - STOP 1: *n* összetett szám, $d(n,m)$ egy osztója
  - STOP 2: *n* összetett szám, de egyetlen osztóját sem találtuk meg
  - STOP 3: *n* valószínűleg prím, a tévedés valószínűsége $< \left(\frac{1}{2}\right)^{100}$
  - az ábra mutatja az algoritmust



- az előző szakaszban látottak szerint a 2. és 3. lépés is elvégezhető polinom időben, így az egész algoritmus lépésszám–igénye is az input hosszának polinomjával becsülhető
- összegezve: a fentiek alapján létezik polinomrendű prímtesztelő algoritmus, ha azonban a beadott szám összetettnek bizonyul, akkor a szorzótényezőit nem tudjuk polinom időben meghatározni
- ha pl. *p* és *q* két egyenként 200 jegyű szám, akkor rövid idő alatt eldönthetjük $n = pq$ -ról, hogy összetett, de mai tudásunk szerint évszázadok alatt sem lehet *n*-ből visszakövetkeztetni *p*-re és *q*-ra

- ez a következő fejezet kulcsgondolata

Nyilvános kulcsú titkosírás

- elképzelhető-e olyan jelszó, amit a rendszer maga sem ismer, és mégis tudja ellenőrizni, hogy mi ismerjük-e?
- válasszunk ki két 200 jegyű *p* és *q* prímszámot és csak az $n = pq$ szorzatukat adjuk meg a gépnek
- ez után rendelkezzünk úgy, hogy annak adhatják ki az adatokat, aki a valamely osztóját mondja meg
- annak ellenőrzése, hogy az adatokért jelentkező személy által mondott *k* szám osztója-e *n*-nek, nyilván gyorsan elvégezhető, de *n*-ből *p* és *q* előállítása mai tudásunk szerint reménytelenül nehéz
- még azt sem kell kérnünk a számítógéptől, hogy az *n* számot (a jelszónkat) tartsa titokban, a konkurencia éppúgy nem tud semmit sem kezdeni az a számmal, mint a számítógép programozója, ez indokolja a "nyilvános kulcs" elnevezést
- ha persze egyszer közöljük a számítógéppel *p* vagy *q* értékét, akkor attól kezdve nem lehetünk biztonságban, a számítógépes adatvédelem ezért egy valamivel bonyolultabb megoldást igényel

Kódolás és dekódolás

- nyilván bármilyen üzenet átalakítható számjegyek sorozatává, vagyis feltehetjük, hogy a titkosítandó majd továbbítandó üzenet mondjuk 400 jegyű számok sorozata

- ha tehát kódolni akarunk egy üzenetet, akkor a kódolást tekinthetjük egy $y = C(x)$ függvénynek, mely a 400 jegyű *x* számhoz egy másik 400 jegyű *y* számot rendel
- e függvény inverzét, az $y = D(x)$ függvényt dekódoló függvénynek nevezzük
- tegyük fel, hogy mindenki nyilvánosságra hozza a saját *C* kódoló függvényét, de titokban tartja a *D* dekódoló függvényt
- ekkor, ha az *i.* személy (a feladó) el akarja küldeni az *x* üzenetet a *j.* személynek (a címzettnek), akkor az általa is hozzáférhető C_j kódolófüggvényt alkalmazva az $y = C_j(x)$ üzenetet küldi el
- a címzett alkalmazza a csak általa ismert D_j dekódoló függvényt és megkapja a $D_j(y) = D_j(C_j(x)) = x$ üzenetet
- a rendszerben részt vevő többi ember számára *y* dekódolhatatlan
- kérdés, hogy lehet olyan $C_1, C_2, \ldots$ kódoló és $D_1, D_2, \ldots$ dekódoló függvényeket készíteni, hogy bármely *x*-re $C_i(x)$ vagy $D_i(x)$ kiszámítása gyorsan elvégezhető legyen, de a C_i ismeretében D_i-re ne lehessen következtetni
- ehhez visszatérünk a prímszámokhoz: tegyük fel, hogy az *i.* résztvevő választ két 200 jegyű prímszámot, jelöljük ezeket p_i-vel és q_i-vel. Legyen $n_i = p_i q_i$
- emlékeztetünk rá, hogy $\varphi(n_i) = (p_i - 1)(q_i - 1)$, jelöljük ezt a mennyiséget m_i-vel
- a résztvevő ezen kívül kiválaszt egy olyan e_i számot is, melyre $1 \le e_i \le n_i$ teljesül és amely relatív prím $(p_i - 1)$-hez is és $(q_i - 1)$-hez is
- végül megoldva egy kongruenciát, meghatározza azt a d_i számot, melyre $e_i d_i \equiv 1 \pmod{m_i}$
- ezután az *i.* résztvevő nyilvánosságra hozza az n_i és e_i számokat, viszont titokban tartja a p_i, q_i, m_i és d_i számokat
- a C_i kódolófüggvény egy *x* üzenethez hozzárendeli azt az $y = C_i(x)$ számot, melyre $y \equiv x^{ei} \pmod{n_i}$, míg a D_i dekódolófüggvény az *y*-hoz annak d_i. hatványát rendeli (mod n_i)
- így $y^{d_i} \equiv x^{e_i d_i} = x^{h m_i + 1} = \left[x^{\phi(n_i)}\right]^h \cdot x \equiv x \pmod{n_i}$
- végül: mindez csak akkor működik, ha *x* relatív prím n_i-hez
- ezt pl. úgy biztosíthatjuk, hogy az üzenetet nem 400, hanem 399 jegyű számsorozatokra bontjuk, majd mindegyik sorozat utolsó elemét úgy választjuk meg, hogy e feltétel teljesüljön
- dekódolás után egyszerűen elhagyjuk az utolsó számjegyet

További trükkök

- a klasszikus titkosírásoknak több gyenge pontja is volt, mindenekelőtt, ha két személy kódolt üzeneteket akart váltani egymással, akkor először meg kellett állapodniuk egymással a kódban (tehát találkozniuk kellett egymással még az üzenetváltás előtt, vagy ha erre nem volt módjuk, akkor biztosítaniuk kellett, hogy amikor az egyik elküldi a másiknak a kódot, az nem kerül illetéktelen kezekbe)
- az előző pontban leírt módszer kiküszöböli ezt a hátrányt
- másik előnye az új módszernek, hogy ha *t* személy akar így levelezni, akkor nem kell $\binom{n}{2}$ féle titkos kódot kitalálni, és mégis bármely üzenet rejtve marad a többi $t-2$ résztvevő előtt
- a régi titkosírások harmadik hátránya, hogy a címzett soha nem tudhatta, hogy tényleg a feladó írt-e neki, vagy „az ellenség” kezébe került a kód, és hamisítványt kap

- az előző pontban leírt módszernek látszólag ugyanez a hátránya (hisz C_j-hez mindenki hozzáfér, nem csak az *i*. résztvevő), azonban a következő trükkel elkerülhetjük ezt a veszélyt:
- az *i*. résztvevő ne *x*-re, hanem $D_i(x)$-re alkalmazza a $w = C_j(z)$ kódolást, majd ezt a *w* „üzenetet" küldi el
- a címzett (tehát a *j*. részvevő) először a csak általa ismert D_j, majd a nyilvánosság számára hozzáférhető C_i függvényt alkalmazza hisz
  $C_i(D_j(w)) = C_i(D_j(C_j(z))) = C_i(z) = C_i(D_i(x)) = x$
- így az üzenetet csak *j* tudja elolvasni, és biztos lehet benne, hogy csak *i* küldhette
- az új módszer negyedik előnyét akkor érthetjük meg, ha nem katonai hírszerzők titkosírásaira gondolunk, hanem egymással konkurens kereskedők, bankárok stb. titkos üzeneteire
- itt ugyanis előfordulhat, hogy *i* rendel valamit *j*-től, majd nem fizet, így *j*-nek bíróság elé kell vinnie az ügyet: be akarja bizonyítani, hogy *i* feladta a rendelést, tehát valamilyen értelemben a kapott *w* üzenetet is, meg annak *x* jelentését is be kell mutatnia, de a bírónak sem akarja megmondani a saját D_j dekódoló eljárását
- ez a látszólag sokkal komplikáltabb feladat is könnyen megoldható: a pert kezdeményező *j* nem csak *w*-t, hanem az $u = D_j(w)$ „félig dekódolt" üzenetet is bemutatja a bírónak
- a bíró kizárólag a nyilvánosság számára is hozzáférhető C_i, C_j kódolási eljárások segítségével ellenőrizheti, hogy
  - $w = C_j(u)$, tehát tényleg *j*-nek jött az üzenet, és hogy
  - $x = C_i(u)$, tehát tényleg *i*-től jött az üzenet

Bizonyítás információközlés nélkül

- a titkosírásokkal kapcsolatos megfontolásaink mind azon alapultak, hogy az összetett számokat nem tudjuk prímtényezőkre bontani
- ha egyszer valaki találna erre a problémára polinomidejű algoritmust, akkor minden ilyen titkos kódot fel tudna törni, márpedig lehet, hogy a probléma polinom időben megoldható
- nincs olyasmi „negatív" eredményünk, mint pl. a Hamilton-kör probléma esetén az **NP–teljes**ség (ami szintén nem bizonyítja ugyan a probléma **P**-n kívüliségét, de legalább erősen valószínűsíti)
- sőt, olyan „pozitív" eredmény is született, hogy egy némiképp hasonló probléma (egész együtthatós polinomok irreduncibilis tényezők szorzatára bontása) éppen hogy polinom időben elvégezhető
- ezért most egy olyan konstrukciót is bemutatunk, ami nem a prímfelbontás nehézségére, hanem a Hamilton–kör létezésének **NP–teljes**ségére épül
- fel fogjuk használni azt, hogy két adott gráf izomorf vagy nem izomorf voltának eldöntésére sem ismeretes polinomrendű algoritmus
- nyilván könnyen tudunk konstruálni egy olyan *G* gráfot, melynek van Hamilton–köre
- ha utána a pontok nevét permutáljuk és így adjuk meg a gráfot, más nem tudja eldönteni, van-e benne Hamilton–kör
- ezt a gráfot adjuk meg titkunk őrzőjének (mondjuk a banknak) azzal, hogy csak annak szolgáltassa ki titkunkat, aki ismeri *G*-nek egy Hamilton–körét
- hogy győzheti meg ezek után a megbízottunk a bankárt, hogy jogosult az információ megszerzésére, anélkül, hogy ezután a bankár is ismerné *G*-nek egy Hamilton–körét?
- megbízottunk mutat egy G_1 gráfot és azt állítja, hogy
  (1) G_1 izomorf *G*-vel, és hogy
  (2) G_1-nek van Hamilton-köre
- ezek után a bankár kérheti, hogy a két állítás egyikét (de csak az egyikét) bizonyítsa be

- ha (1)-et kell bebizonyítani, akkor megbízottunk megad G és G_1 ponthalmaza között egy kölcsönösen egyértelmű és szomszédosságtartó leképezést
- ha (2)-t kell bebizonyítani, egyszerűen megad egy Hamilton–kört
- az első esetben a bankár nem jut semmilyen új információhoz, csak az általa már úgyis ismert G-nek egy izomorf G_1 leírását is látni fogja
- a második esetben sem jut semmilyen információhoz: lát ugyan egy G_1 gráfot, melynek van Hamilton–köre, de mivel nem tudja eldönteni, hogy G és G_1 izomorfak-e, változatlanul nem tud G-hez Hamilton–kört rajzolni
- ennek ellenére annak a valószínűsége, hogy megbízottunk „blöffölt", vagyis nem ismeri G-nek egyetlen Hamilton–körét sem, legfeljebb $\frac{1}{2}$ lehet
- ő ugyanis nem tudhatja előre, hogy a bankár (1) vagy (2) bizonyítását fogja kérni
- így akár G-t rajzolja át G_1-gyé anélkül, hogy ismerné egy Hamilton–körét, akár egy Hamilton–körrel rendelkező G_1-et rajzol anélkül, hogy tudná, izomorf-e G-vel, $\frac{1}{2}$ valószínűséggel lelepleződnék
- lehet mondani, persze, hogy a $\frac{1}{2}$ valószínűség túl nagy, de ha a bankár mondjuk százszor kéri a „jelszót", vagyis száz darab G_1 , G_2, ..., G_{100} gráfot, és mindegyiknél egymástól függetlenül dönti el, hogy az (1) vagy a (2) állítás bizonyítását kéri, akkor vagy mind a száz bizonyítás kielégítő, vagy van legalább egy olyan állítás, amit a titokért jelentkező nem tud bebizonyítani
- utóbbi esetben biztos, hogy nem ismeri a jelszót, előbbi esetben $\left(\frac{1}{2}\right)^{100}$ -nál kisebb a „blöffölés" valószínűsége (tehát lényegében zérus)
- ezt a kockázatot már vállalhatja a bankár
- ha valaki tényleg meg akarna valósítani egy ilyen rendszert, akkor G megválasztása komoly problémát vethet fel: ha túl sok, vagy túl kevés élű gráfot választanánk, akkor esetleg könnyű lenne benne Hamilton–kört találni