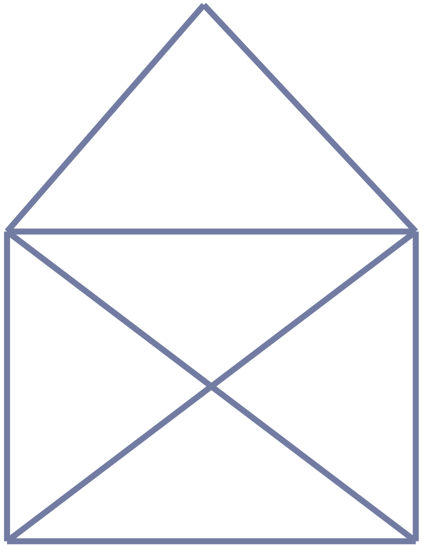
Algoritmusok, problémaosztályok

Gráfok

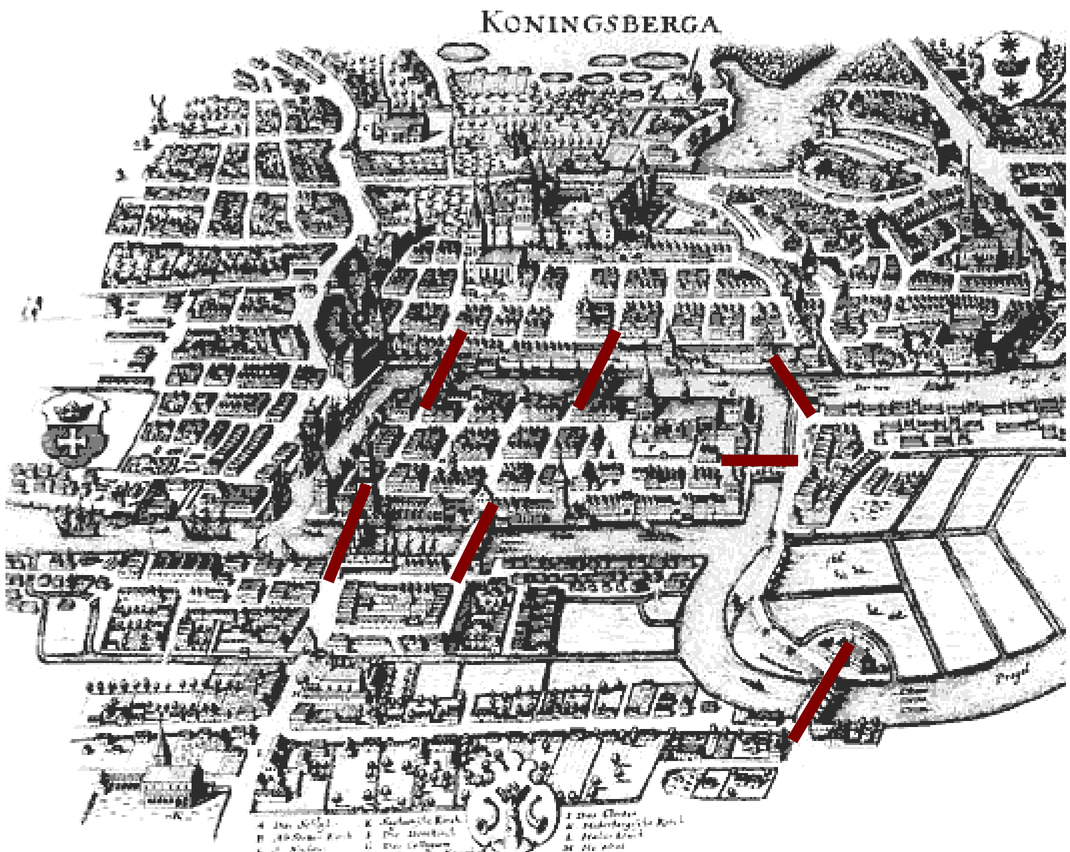
Bevezető



- Hogy lehet ezt az ábrát úgy lerajzolni, hogy minden vonalon csak egyszer menjünk végig?

Königsberg hídjai

- Hogy lehet Königsberg minden hídján úgy végigmenni, hogy mindegyiken csak egyszer menjünk végig?



Königsberg hídjai

- Euler oldotta meg
  - Nem lényeges az út a szárazföldön, csak a hidak
  - Elvont modell:
    - Szárazföld: csomópont
    - Hidak: utak a csomópontok között – élek
    - Nem számít a csomópontok elhelyezése és az élek formája

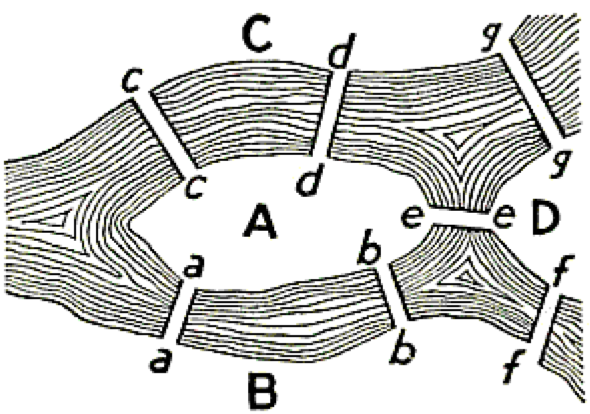
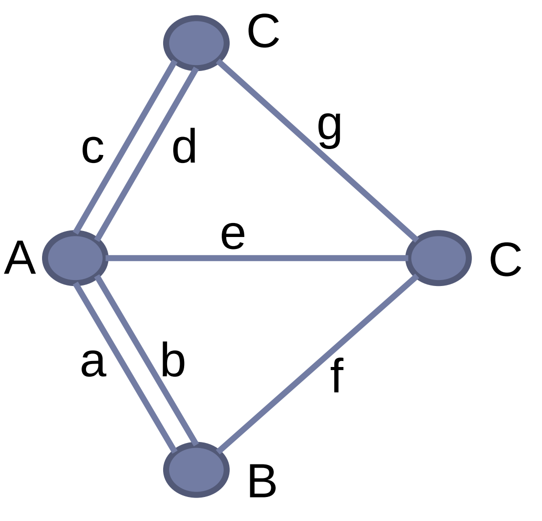


FIGURE 98. *Geographic Map: The Königsberg Bridges.*



Megoldás

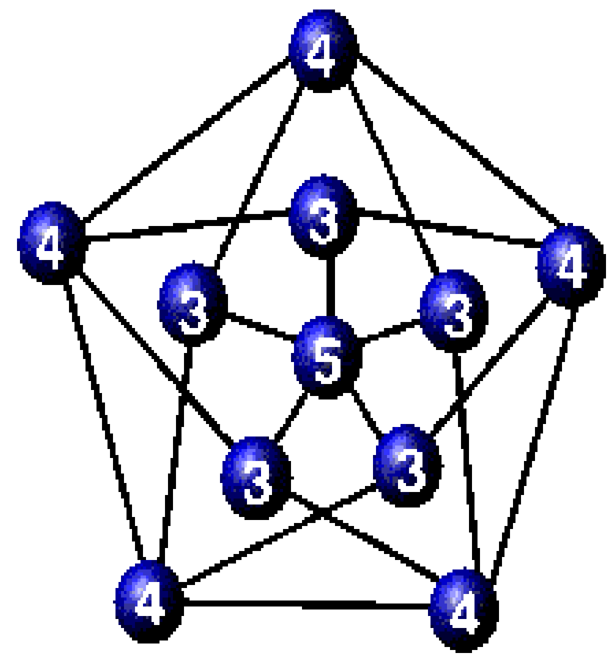
- Ha egy pontba beérünk, onnan ki is kell menni – kivéve a kezdő és végpontot
- Tehát kettő kivételével (kezdő és végpont) minden csomópont páros számú éllel kell rendelkezzen (csúcspont fokszáma)
- Ha ugyanoda szeretnénk visszaérni, akkor minden csúcspont páros fokszámú kell legyen.
- Königsberg esetében mind a négy csomópont páratlan fokszámú

Gráfok

- A **gráf** egy olyan struktúra, ami **csúcsokból** vagy szögpontokból és **élekből** áll, minden él két csúcs között fut.
- A csúcsok tartalmazzák az objektumokat (adatokat), az élek pedig meghatározzák a csúcsok közti összefüggéseket.
- Legtöbbször **egyszerű gráfokkal** foglalkozunk, azaz olyanokkal, amelyekben nincs **hurokél** (egy csúcsot önmagával összekötő él) és nincsenek **párhuzamos élek** sem, tehát azonos pontok között haladó különböző élek.

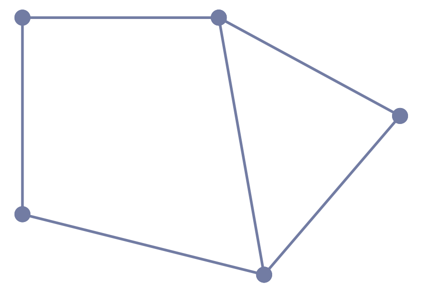
Gráfok

- Egy **csúcspont fokszáma** a rá illeszkedő élek száma. Ha ez nulla, tehát az adott csúcsra nem illeszkedik él, akkor a csúcs **izolált**. Véges gráfok esetén a fokszámokat összeadva páros számot kapunk, hiszen ekkor minden élt kétszer számoltunk. Ezért a páratlan fokú pontok száma mindig páros.
- Ha a fokszám minden csúcsra azonos, a gráf **reguláris**. Ha ez a közös fokszám k, akkor a gráf k-adfokú reguláris vagy **k-reguláris**.
- A mellékelt ábrán minden csúcsba beírtuk a csúcs fokszámát

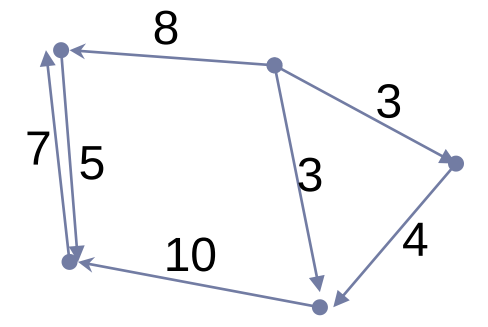


Élek

- Az élek lehetnek
  - Kétirányúak
  - Irányítottak



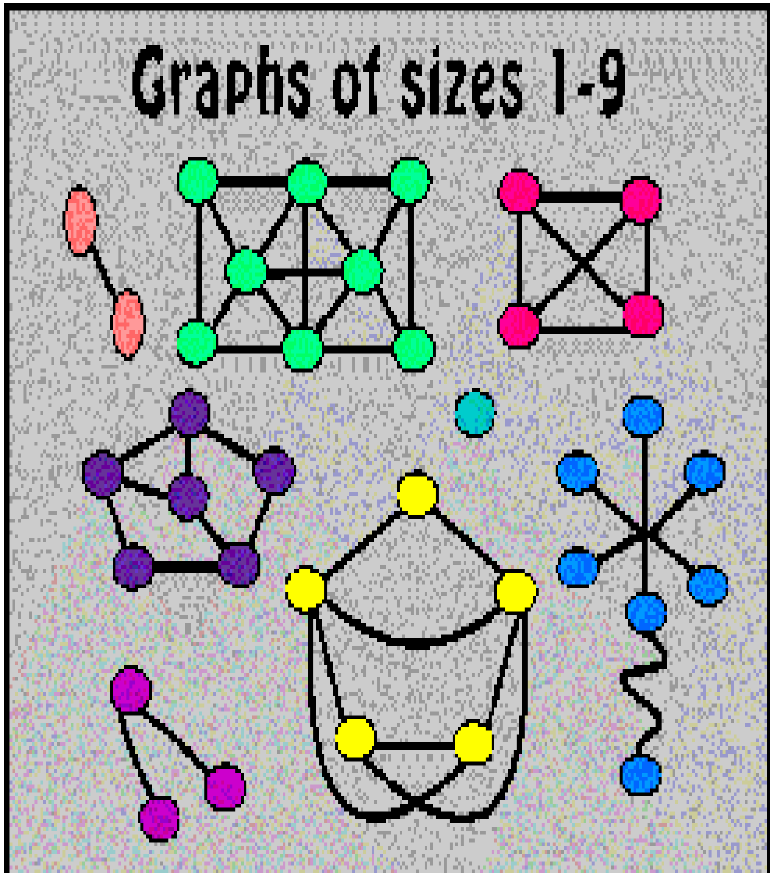
  - Egyszerűek
  - Súlyozottak



- Különleges élek (pl: hurokél)

A gráf mérete

- A gráf mérete megegyezik a gráfot alkotó csúcsok számával



Utak

- Az **út** élek egymáshoz csatlakozó sorozata, amely egy csúcsot legfeljebb egyszer tartalmaz.
- A **kör** élek egymáshoz csatlakozó sorozata, ami záródik, tehát az utolsó és az első élnek van közös végpontja, és nincs ismétlődő csúcs.
- Ha csak élek ismétlődését zárjuk ki, akkor **ciklusról** beszélünk.
- Az út hossza egyenlő az érintett csúcsok számával

Euler-gráfok

- **Euler-út**: Olyan út amely minden élen végigmegy. Az Euler út egy csúcsot többször is érinthet.
- Azt az Euler-utat amely ugyanabba a pontban végződik ahonnan elindult, **Euler-körnek** nevezzük
- Az olyan gráfokat, amelyben léteznek Euler-körök, **Euler-gráfnak** nevezzük

- Egy gráf akkor Euler-gráf, ha nincsenek izolált részei és minden csomópontja páros fokszámú

Hamiltom-gráfok

- **Hamilton-utak**: azok az utak, amelyek minden pontot pontosan egyszer érintenek. A Hailton-út nem kell minden élet bejárjon.
- Az olyan Hamilton-útat amely a kezdőpontban végződik **Hamilton körnek** nevezzük
- Az olyan gráfokat, amelyben léteznek Hamilton-körök, **Hamilton-gráfnak** nevezzük

- Ha egy n csomópontból álló gráf minden csomópontjának a fokszáma nagyobb mint n/2, akkor a gráf Hamilton-gráf.

Két pont távolsága

- Két pont távolsága egyenlő a köztük levő legrövidebb út hosszával.
- Vigyázat, két pont között több út is lehetséges – a távolság ezek közül a legrövidebb.
- Egy gráf átmérője: A leghosszabb távolság egy gráfban

Algráfok, részfák

- Az algráfok olyan gráfok, amelyek tartalmazzák egy gráf csúcsainak egy részét, és az összes élet, amely ezeket összeköti (pl: egy várostérkép az országtérkép részgráfja)

- A részfa egy olyan gráf, amely tartalmazza az eredeti gráf összes csúcsát, viszont nem tartalmaz köröket (tehát elhagyjuk az összes olyan élet, amely kört alkotna). Pl. az optimális utak egy adott pontból.

Kapcsolatok a gráfokban

- Kapcsolt gráfok azok a gráfok, amelyben bármely két pont közt van út
- Kapcsolt komponens egy olyan kapcsolt algráf, amelynek nincs kapcsolata a gráf többi részével

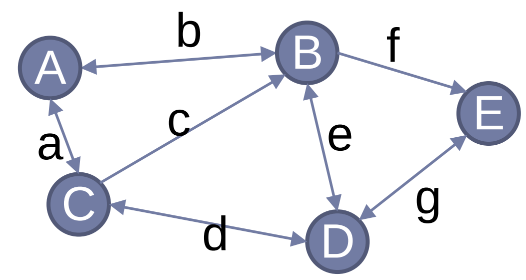
- Komplett gráfnak nevezzük az olyan gráfot, amelynek minden pontja össze van kötve minden ponttal
- A komplett gráf átmérője 1
- A komplett gráfhoz már nem lehet éleket adni.

Szociális hálózatok

- Karinthy Frigyes fogalmazta meg a “kis világ” elméletét a “Láncszemek” c. novellában: Bárkitől el lehet jutni bárkihez öt ismerősön keresztül (“hatlépésnyi távolság”).
- Egy szociális hálózat jellemzői:
  - A pontok fokszáma
    - Hálózati központok
  - Csoportosulások (komplett algráfok)
  - Legrövidebb út megtalálása
  - Az élek különböző típusai

Gráfok ábrázolása

- Geometriai ábrázolás
- Lista
  - minden csúcsnak felsoroljuk a szomszédait
  - a listákat egymás után írva, csillaggal elválasztva, a végére két csillagot téve

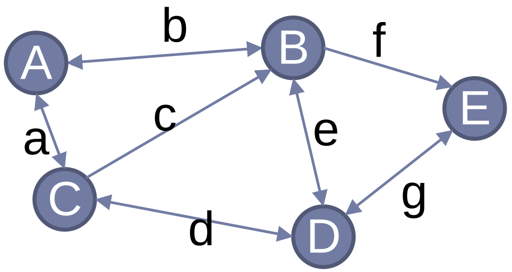


Lista
A→B→C
B→A→D→E
C→A→B→D
D→B→C→E
E→D

Lista tömbbel
B C * A D E * A B D * B C E * D * *

Gráfok ábrázolása

- Geometriai ábrázolás
- Lista
- Mátrix
  - Csúcsmátrix
  - Pont-él mátrix



Csúcsmátrix

	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	1
C	1	1	0	1	0
D	0	1	1	0	1
E	0	0	0	0	1

Pont-él mátrix

	a	b	c	d	e	f	g
A	1	1	0	0	0	0	0
B	0	1	1	0	1	-1	0
C	1	0	-1	1	0	0	0
D	0	0	0	1	1	0	1
E	0	0	0	0	0	1	1

Példák gráfokra

- Számítógépes hálózatok
- Egyéb hálózatok (elektromos, víz, stb)
- Ismeretségi hálók
- Térképek
- Forgalom-modellezés
- Projektek
- Órarend, menetrend…
- Adatbázisok

Gráfalgoritmusok

- Algoritmikus problémák megoldása során gyakran van szükség dolgok közötti bináris kapcsolatok kezelésére, erre modellként szolgálhatnak a gráfok.
- Megfeleltetés: a gráf csúcsai az objektumok, a gráf élei az objektumok közötti kapcsolatoknak felelnek meg. (Pl.: Baranya megye települései, a településeket összekötő utak)
- A gráf mint modellezési eszköz népszerű a számítástudományban egyszerűsége, kezelhetősége, kifejező képességének tágassága miatt.

- **Hálós megközelítés**: az egyik legjelentősebb adatmodellező irányzat; irányított gráfokat alkalmaz. Gráf az egész világ. Gráfoknak tekinthetők a belső memóriás adatkezelés listaszerkezetei is, ahol az adatelemek közötti mutatók jelentik az éleket. Ebben a témakörben gyakran előforduló algoritmusok az elérhetőséggel, összefüggőséggel, bejárhatósággal kapcsolatosak.

- Mi csak az alapfogalmakkal és néhány, az alkalmazásokban gyakran felmerülő feladattal foglalkozunk.

Gráfalgoritmusok

- Egy G gráf két halmazból áll:
  - a csúcsok (vagy pontok) **V** halmazából, mely egy véges, nem üres halmaz
  - az élek **E** halmazából, amelynek elemei bizonyos V-beli párok.
  - A halmazok elemszáma: **n = |V|** és **e = |E|**
  - A gráf jelölése: **G = (V, E)**

Alapműveletek gráfokban

- `szomszéd(G,x,y)` – igaz, ha x és y csúcs össze van kötve a G gráfban
- `szomszédok(G,x)` – egy listában visszaadja az x csúcs szomszédait
- `összeköt(G,x,y)` – összeköti az x és y pontokat egy éllel (ha ez nem létezik még).
- `töröl(G,x,y)` – törli az élet az x és y pontok közt
- `csúcs_érték(G,x)` – lekéri az adott csúcs értékét
- `csúcs_érték_állít(G,x,a)` – megváltoztatja az adott csúcs értékét
- `él_érték(G,x)` – lekéri az adott él értékét
- `él_érték_állít(G,x,a)` – megváltoztatja az adott él értékét

Gráfalgoritmusok

- Bejárások
  - szélességi
  - mélységi
- Minimális költségű út (Dijkstra, Floyd algoritmusok)
- Összefüggő komponensek, komplett részgráfok
- Párosítások
- Kritikus út projektekben
- Áramlás-sűrűség

Szélességi bejárás

- Egy x pontból kiindulva kell bejárni a gráfot.
- Előbb végigmegyünk a pont összes szomszédján, utána ezeknek a szomszédain (amelyeken még nem mentünk végig) amíg minden pontot érintünk.
- Szükséges két tömb:
  - S (sor) – ebbe tároljuk a bejárt elemeket (E,V)
  - Bejárt – egy adott eleme 1, ha azt a csúcsot bejártuk
- **Pszeudokód**

```
push(S,x); bejart[x]=1;
amig(nem ures a sor)
  pop(S,y);
  z=szomszédok(y);
  minden z-re ahol bejart(z)=0
        push(S,z);
        bejart[z]=1;
Ki:S;
```

Mélységi bejárás

- Ebben az esetben minden pont bejárásakor megpróbálunk továbblépni ameddig csak lehet.
- Ezután lépünk az adott pont következő szomszédjára
- 2 tömb: V (verem), bejárt.
- **Pszeudokód**:

```
Push(V,x);
amíg(nem ures a verem)
  pop(V,y);
  Ki:y;
  bejart[y]=1;
  z=szomszédok(y);
  minden z-re ahol bejart[z]=0
        push(V,z);
```

Roy-Floyd algoritmusa

- Meghatározza a legrövidebb utakat egy gráfban bármely két pont között
- N: a gráf csomópontjainak száma
- C: az utak mátrixa.
  - C[i,j]: az i és j pont közti legrövidebb út hossza (ha nincs út akkor ∞)
- **Pszeudokód**

```
ciklus k=1-től N-ig
    ciklus i=1-től N-ig (i!=k)
        ciklus j=1-től N-ig (j!=k)
            C[i,j]=min(C[i,j],C[i,k]+C[k,j]
```

Dijkstra algoritmusa

- Meghatározza a legrövidebb utakat egy gráfban egy adott pontból kiindulva.
- A legrövidebb utak fáját építi fel

- Lépések:
  - Minden csúcspontban a legrövidebb távolságot tároljuk az x0 csúcstól
  - x0 szomszédjainak értéket adunk
  - A legközelebbi csúcs szomszédjainak is értéket adunk: Ha már van értéke, akkor a kisebb értéket veszi fel az új és a régi érték közül.
  - Az előző lépést ismételjük a következő ponttal, amíg be nem járjuk a gráfot

GPS szoftverek működése

- Gráfok:
  - Csomópontok: útkereszteződések
  - Élek: utak. Ezek legtöbbször irányítottak és több távolságot tárolnak (táv, idő, ár).
- Az útkeresési algoritmusok nagyon hasonlítanak az eddig bemutatottakhoz
- Az útkeresés két fázisban történik:
  - Előfeldolgozás: célja egy majdnem teljes gráf létrehozása. Ez tartalmazza egyrészt az optimális áthaladást a városokon, valamint a főbb csúcspontok közti összeköttetéseket. Ez a lépés több napot is igénybe vehet.
  - Hierarchikus hálózat kialakítása: hosszú távon a főútvonalakat fogja csak használni.
  - Útkeresés: Itt nem kell az egész gráfot bejárni, elég egy pár lépés mélységbe menni