Molnár Martin – BME-VIK
2017.10.25.

Számítástudomány alapjai I. ZH segédlet

Leszámlálási alapfeladatok

	ismétlés nélküli	ismétléses
permutáció	$n!$ n futó beérkezésének sorrendje	$\frac{n!}{k_1!k_2!\ldots k_r!}$ n golyót ennyiféleképpen állíthatunk sorba, ha $k_1, k_2, \ldots k_r$ db külön-külön egyszínű
variáció	$\frac{n!}{(n-k)!}$ n futó beérkezésének sorrendje ha csak az első k helyet tekintjük	l^k l darab betűből készíthető k hosszú szavak száma (egy-egy betű többször is felhasználható)
kombináció	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$ n golyóból kiválasztunk k darabot (nem számít a kiválasztás sorrendje)	$\binom{k+l-1}{l}$ k féle sütiből (sok van belőlük) hazaviszünk l -et, ennyiféleképpen tehetjük meg *(Extra tananyag)*

Gráfelméleti alapok

G egyszerű véges gráf: G=(V,E) ahol V a csúcsok, E pedig az élek halmaza.
Speciális élek lehetnek: párhuzamos élek, hurokélek, irányított élek.
Egy G irányított gráfban a csúcsok kifokának összege megegyezik a befokok összegével; ez az összeg a gráf éleinek számát adja.
Irányítatlan gráf esetében a fokszámok összege az élszám kétszerese.

Egyszerű véges G=(V,E) **gráf komplementere** G', ha G' gráfban pontosan ott fut él, ahol G-ben nem, és a csúcshalmazaik megegyeznek.
Teljes gráfot K-val, kört C-vel, utat P-vel jelölünk.
G és H gráf akkor **izomorfak**, ha csúcsaik megszámozhatóak úgy 1-től n-ig, hogy az azonos számpárok között ugyanannyi él fut G-ben mint H-ban.
A **séta** olyan élsorozat, amelyben él nem ismétlődhet.
Az **út** olyan élsorozat, melyben sem él, sem csúcs nem ismétlődhet. Egy út nyílt ha kezdő- és végpontja különböző; zárt, ha megegyezik.
G gráf **összefüggő**, ha bármely két csúcs között vezet út.
K egy **komponense** G-nek, ha K bármely két csúcsa közt vezet G-ben út, de K-n belüli és K-n kívüli csúcsok közt nem létezik út G-ben.

H gráf **feszített részgráfja** G-nek, ha H megkapható csúcstörléssel.
H gráf **feszítő részgráfja** G-nek, ha H megkapható éltörléssel.
H gráf **részgráfja** G-nek, ha H megkapható él- és csúcstörléssel.
G=(V,E) gráf **erdő**, ha G körmentes. G gráf **fa**, ha erdő és összefüggő.
Egy n csúcsú **fa éleinek száma** n-1.
Ha G egy n csúcsú erdő, és k komponense van, akkor éleinek száma n-k.
Tetszőleges n csúcsú G gráfban az alábbi három tulajdonság közül bármelyik kettő maga után vonja a harmadikat:

1. G gráf körmentes,
2. G gráf összefüggő,
3. G gráf éleinek száma n-1.

Ha F gráf fa, és az e él éle F-nek, akkor a G-e gráfnak pontosan két komponense van.
Egy F fába tetszőleges élt behúzva pontosan egy kör jön létre.
F fában bármely u,v csúcsok között pontosan egy út létezik.
Egy gráfban v csúcs levél, ha fokszáma pontosan 1.
Ha F fa, és legalább 2 csúcsa van, akkor F-nek van legalább 2 levele.
F a G **feszítőfája**, ha F fa, és F a G feszítő részgráfja.
Ha G-nek létezik feszítőfája, akkor G összefüggő.
G gráf akkor és csakis akkor (iff) összefüggő, ha G-nek létezik feszítőfája.

Algoritmusok

Kruskal: ha egy tetszőleges G gráfban értelmezett költségfüggvény súlyokat rendel az élekhez, akkor mohó módon állíthatunk elő minimális költségű (súlyú) feszítőfát. Mindig a lehető legkisebb költségű élet húzzuk be a gráfba, mindaddig, amíg kör nem keletkezik. Az így keletkezett fa a G gráf minimális súlyú feszítőfája.

BFS: G gráfban kiindulunk tetszőleges (A) pontból. Ha A-ból vezet ét még eléretlen csúcsba, akkor azt behúzzuk; ezek a csúcsok elértté válnak. Ha A-ból nem tudunk több eléretlen csúcsba menni, akkor A befejezetté válik. Ekkor az A után először elért csúcsból folytatjuk ugyanezt. Több eléretlen csúcs esetén abc-sorrendben választunk eléretlen csúcsot. A bejárás után elérési és befejezési sorrendeket kapunk (amik egyébként megegyeznek). A BFS bejárás fáját úgy kapjuk meg, hogy minden elért csúcsra felírjuk melyik él által értük el. Ezek a faélek alkotják a BFS-fát. BFS bejárás után csak fa-, kereszt-, esetleg visszaél lehet, előreél nincsen; át kellene ugrania egy faélt. A BFS-fa egy gyökérpontból minden más csúcsba a legrövidebb utakat tartalmazza, a BFS (szélességi keresés = Breadth First Search) által megkapott BFS-fa a legrövidebb utak fája. Lépésszáma legfeljebb konstans*(n+m), ahol n a csúcsok, m az élek száma.

Dijkstra: Irányított G gráfban r gyökérpontból indulva szeretnénk elérni az összes többi csúcsot a lehető legrövidebb úton. G gráfon legyen értelmezett egy hosszfüggvény (l), amely minden élre egy l hosszt határoz meg. A kiindulási pont (gyökérpont) távolsága legyen 0, a többi (még eléretlen) csúcsé kezdetben végtelen (mert nem értük el). Sorban feljegyezzük, hogy melyik csúcsot milyen távolságban (vagy milyen költség árán) érjük el. Az A-ból vezető éleket kezdetben mindenképpen behúzzuk a csúcsokba. Ezen csúcsokhoz feljegyezzük, hogy az A-ból értük el őket, és azt is, hogy milyen messze vannak tőle. Ekkor A-ból már nem tudunk mit csinálni, befejezzük. Az elért csúcsok közül a legolcsóbbat (legközelebbit) kiválasztjuk, és megnézzük hova vezet él. Abban az esetben, ha olyan csúcsba vezet él, melyet már A-ból elértünk, megpróbálunk élmenti javítást végezni. Ha C pontot A-ból 90 messze tudtunk elérni, de B távolsága A-tól 30, illetve C 40 messze van B-től, akkor a C-be vezető teljes út hossza így csak 70 lesz, ha B-n keresztül érjük azt el. Ezt az élmenti javítást minden lehetséges esetben megtesszük az algoritmus futtatásakor, és az előzőekhez hasonlóan feljegyezzük, hogy milyen csúcson keresztül értük el, és mennyi összköltség fejében. Az algoritmust mindig a lehető legolcsóbban elért csúcsból folytatjuk, és minden iterációban felírjuk egymás alá sorokban az összes csúcs adatát, állapotát. Ha helyesen futtatjuk a Dijkstra algoritmust, akkor végül egy élsúlyozott legrövidebb utak fáját kapunk (ezen éleken keresztül állítottuk be a végső költségeket vagy távolságokat). Lépésszáma legfeljebb konstans*n^2.

Ford: Szinte megegyezik a Dijkstra algoritmussal, de megengedünk negatív élsúlyokat is. A feltétel mindössze annyi, hogy negatív összsúlyú kör nem lehet G gráfban.

Floyd: ez az algoritmus inputként egy irányított G gráfot kap, outputként pedig meghatározza bármely két csúcs közti út hosszát. Alkalmazásakor egy gráf csúcsaiból mátrixot alkotunk (n darb csúcsból n*n-es mátrix); a szimmetriatengely természetesen mindenhol 0, az egymásból közvetlen elért csúcsok közti hosszokat beírjuk, míg a többi helyre végtelen kerül. Ez a nulladik lépés. Utána az első lépés, hogy az első oszlopot és sort letakarjuk a mátrixban, úgymond ezek érintetlenek maradnak, ahogy a szimmetriatengely is. A változatlanul leírt értékek metszéspontjai az összegek lesznek, és ha az összeg kisebb, mint az amúgy ott lévő, akkor lecseréljük. A lépésszám max konstans*n^3.

Irányítatlan G gráfban csökkenő szélesség szerint futtatott Kruskal algoritmus legszélesebb utak fáját eredményezi.
G gráf irányított aciklikus (körmentes) gráf **(DAG**: directed acyclic graph), ha nincsen benne kör.
Ezzel egyenértékűek: G DAG; G csúcsainak létezik topologikus sorrendje; DFS után nincsen visszaél.

DFS: G=(E,V) gráfban legyen értelmezett l hosszfüggvény, és keressük a gráfban a leghosszabb utat. A DFS (Depth First Search) bejárás magyarul mélységi keresést vagy bejárást jelent. A bejárás során a következő csúcsot mindig a lehető legkésőbb elért csúcsból érjük el (abc sorrend számíthat). Outputként elérési és befejezési sorrendet kapunk. Ha egy csúcsból nem tudok tovább menni (nem tudok még el nem ért élt elérni), akkor a csúcs befejezetté válik, és az előzőből próbálkozunk. DFS futtatása által kapott befejezési sorrend egy fordított topológiai sorrend. DFS után irányítatlan gráfban nem lehet keresztél. Az algoritmus lépésszáma legfeljebb konstans*(n+m).

PERT: A projekt menedzselési probléma adott tevékenységek által alkotott projekt ütemezését segít meghatározni. Egy G gráfban határozzunk meg egy topológiai sorrendet (DFS-sel a befejezési sorrend megfordítva vagy források folyamatos elhagyásával), majd rajzoljuk újra a gráfot a topológiai sorrend segítségével. Ezután határozzuk meg a legkorábbi kezdési időpontokat úgy, hogy az előzőt (élen futó érték) mindig hozzáadjuk az aktuálishoz; több lehetséges összeg esetén mindig a maximálisat vegyük. Ezután a projekt utolsó tevékenységétől visszafele haladva megnézem, hogy minek a hatására kapta a kezdési időpontját. Ezeket az éleket bejelölve megkapom a kritikus utat, amely a nyelőbe vezető leghosszabb él. Az ezen a kritikus úton elhelyezkedő tevékenységeket kritikus tevékenységeknek nevezzük: ezek csúszása az egész projekt csúszásához vezetnek. A nem kritikus úton lévő tevénységek is okozhatnak csúszást, ha a késedelem elég nagy.

Utak, séták, körök, körséták

Egy G=(E,V) irányított gráfnak létezik **Euler-körsétája**, ha van G-ben olyan séta (körséta), amely G minden élét tartalmazza. Euler-körsétáról beszélhetünk, ha a séta kezdő és végpontja azonos; **Euler-sétáról** ha ezek különbözőek.
Ha G-nek van Eksja, akkor G az izolált pontoktól eltekintve összefüggő.
Ha G-ben van Es, akkor lehet benne Eks.
Euler-körséta létezik G-ben, ha: (1) G izolált pontjaitól eltekintve összefüggő, és (2) G-ben minden fokszám páros.
Euler-séta létezik G-ben, ha: (1) G izolált pontjaitól eltekintve összefüggő, és (2)' legfeljebb 2 páratlan fokú csúcs van (ezek a kezdő és a végpont, és csak 1 páratlan nem lehet).
Euler-körsétát úgy csinálhatunk egyszerűen, ha egy irányított G gráfot éldiszjunkt körökre bontunk fel úgy, hogy minden csúcsban összekötünk tetszőleges élpárokat. Ha nem egy körsétát kaptunk, akkor egy csúcsban összefűzzük őket, és mindezeket addig ismételjük, míg pontosan egy körsétát nem kapunk.

Egy G=(E,V) gráf **Hamilton-köre** egy olyan kör, amely G minden csúcsát tartalmazza. **Hamilton-útról** akkor beszélünk, ha nem érünk vissza a kiindulási pontba.
Szükséges feltétel: ha G-nek létezik Hamilton-köre (vagy útja), akkor bármely k-ra teljesülnie kell annak, hogy G-ből k db csúcs törlésével legfeljebb (k+1) komponens jön létre.
Elégséges feltétel:

(1) **Dirac-tétel**: n csúcsú G gráf esetén ha minden csúcs fokszáma legalább n/2, akkor G-nek létezik Hamilton-köre.
(2) **Ore-tétel**: egy legalább 3 csúcsú egyszerű G gráfban ha u és v között nem fut él, és u és v fokszámainak összege (d(u) + d(v)) legalább n, akkor G-ben van Hamilton-kör.

A Dirac-feltételből következik > az Ore-feltétel, az Ore-tételből kötvetkezik > a Dirac-tétel.