Adatszerkezetek és algoritmusok

Gráfok

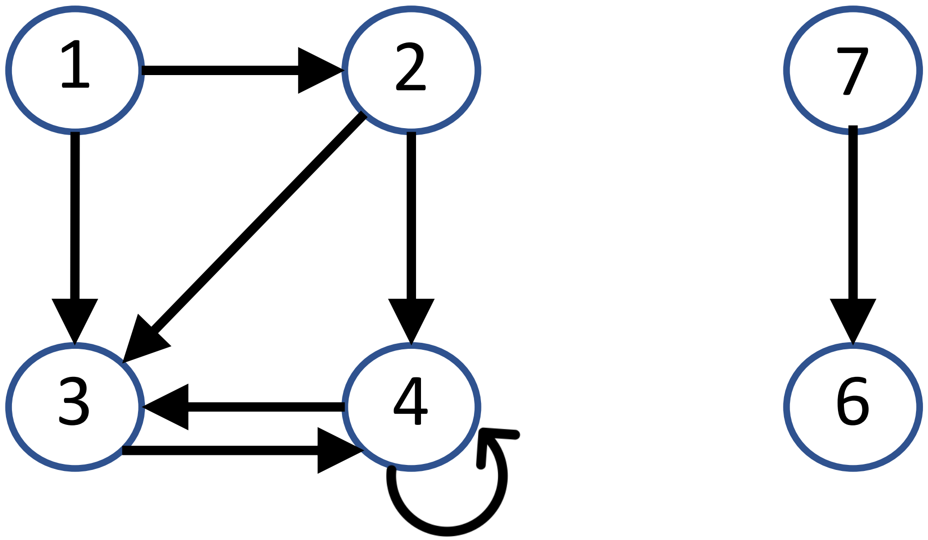
Dr. Fazekas Attila

A tananyag elkészítését az EFOP-3.4.3-16-2016-00021 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

DEBRECENI EGYETEM

Alapfogalmak

Gráf, csúcs, él, irányított gráf, hurokél, csúcsok száma, élek száma.



DEBRECENI EGYETEM

Alapfogalmak (folyt.)

- **Irányítás nélküli gráf**.
- **Súlyozott gráf** (súlyfüggvény értelmezve az éleken).
- **Egyszerű gráf** (olyan gráf, amelyben nincs hurokél, illetve többszörös él).
- **Szomszéd/rákövetkező csúcs**.
- **Út**, **kör**.
- **Egyszerű út** (olyan út, amely körmentes).

Alapfogalmak (folyt.)

- **Egyszerű kör** (egy olyan kör, amelyben nincs belső kör, azaz páronként különböző csúcsokból áll).
- **Körmentes gráf**.
- **Fokszám**, **kimenő fokszám**, **bemenő fokszám**.
- **Összefüggő gráf** (egy irányítás nélküli gráf összefüggő akkor, és csak akkor, ha bármely két csúcs összeköthető úttal).

Alapfogalmak (folyt.)

- **Erősen összefüggő gráf**.
- **Teljes gráf** (egy olyan irányítás nélküli gráf, amelynek bármely két csúcsa szomszédos).
- **Páros gráf** (egy olyan irányítás nélküli gráf, amelynek csúcsai két diszjunkt halmazra bonthatóak, és él csak a két különböző halmaz csúcsai között mehet, azonos halmazban lévő csúcsokat azonban nem köthet össze).
- **Erdő** egy körmentes, irányítás nélküli gráf. **Fa** egy összefüggő erdő.

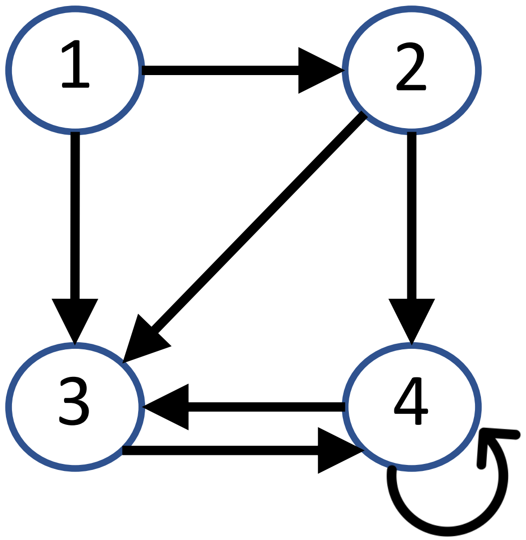
DEBRECENI EGYETEM

Gráfok ábrázolása

- Ha adott egy gráf *n* db csúccsal, akkor a gráfot egy *nxn*-es mátrixban ábrázolhatjuk, ahol az oszlopokat és a sorokat rendre a csúcsokkal indexeljük. Egy mezőben akkor van 1-es, ha a megadott csúcsok között van él, különben 0. Ezt a mátrixot **szomszédsági mátrix**nak vagy **csúcsmátrix**nak nevezzük.

DEBRECENI EGYETEM

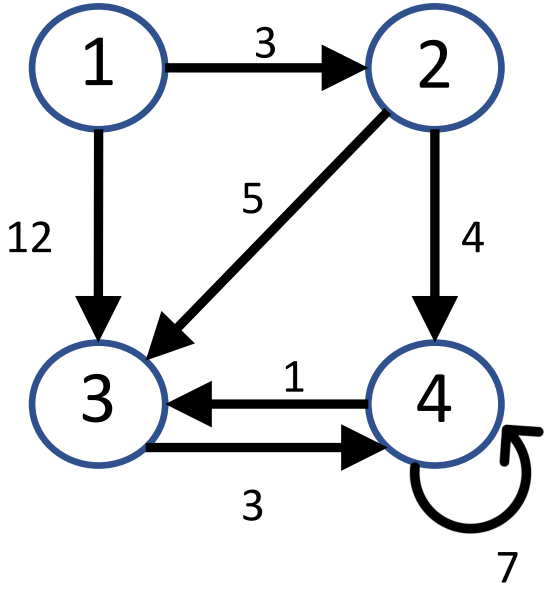
Példa – Gráfok ábrázolása



	1	2	3	4
1	0	1	1	0
2	0	0	1	1
3	0	0	0	1
4	0	0	1	1

Gráfok ábrázolása (folyt.)

Súlyozott gráfok esetén a mátrixban a 0 és 1 értékek helyett az élsúlyokat tároljuk. **Végtelen élsúly** jelzi az él hiányát két csúcs között.



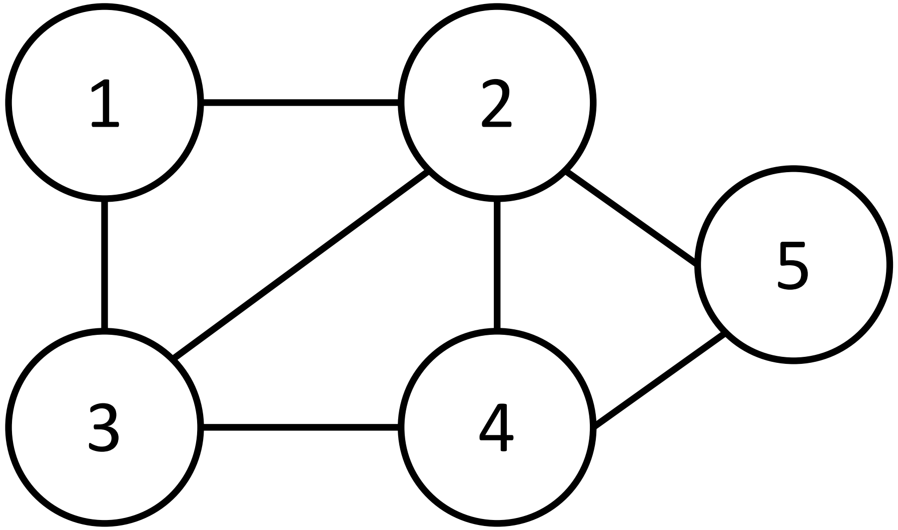
	1	2	3	4
1	∞	3	12	∞
2	∞	∞	5	4
3	∞	∞	∞	3
4	∞	∞	1	7

DEBRECENI EGYETEM

Éllistás ábrázolás

- Éllistás reprezentációban a gráf minden csúcsához egy listát rendelünk. Ebben a listában tarjuk nyilván az adott csúcsból kimenő éleket.
- Ritka és normális mátrixok esetén érdemes alkalmazni ezt a reprezentációt. Sűrű mátrixok esetén sok a járulékosan tárolandó adat, így ott a csúcsmátrix használata célszerűbb.

Példa – Éllistás ábrázolás



1 → 2 → 3
2 → 3 → 4 → 5 → 1
3 → 1 → 2 → 4
4 → 3 → 2 → 5
5 → 2 → 4

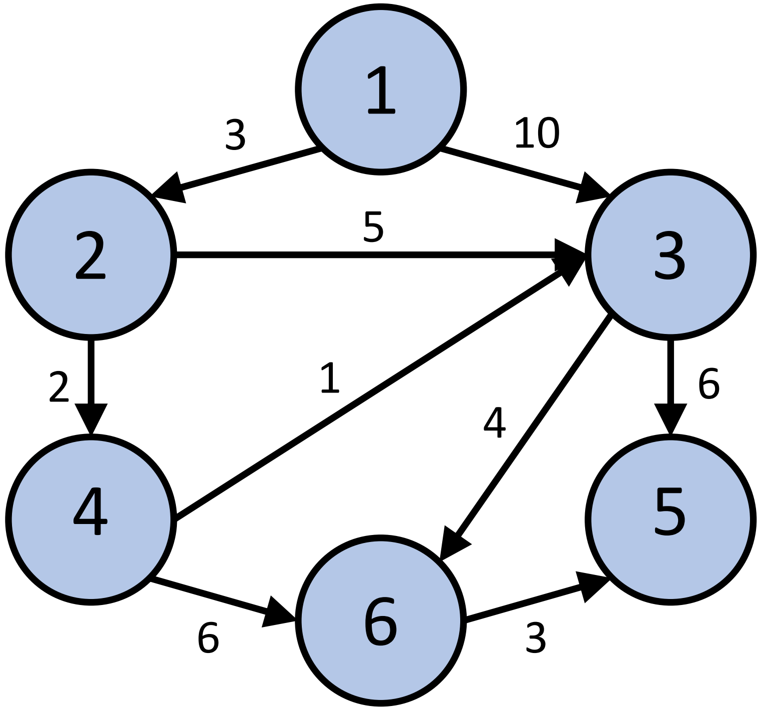
Legrövidebb út - Dijkstra algoritmus

- Egy súlyozott gráfban adott két csúcs között keressük a legrövidebb súlyösszegű utat.
- Két tömböt fogunk használni, a d[1..n] és P[1..n] tömbökben a távolságot és a megelőző csúcsot tartjuk nyilván.
- A KÉSZ halmazba rakjuk azokat a csúcsokat, amelyekhez már ismerjük a legrövidebb utat.
- Ezen kívül, használunk egy minimum választó elsőbbségi (prioritásos) sort (minQ), amelyben a csúcsokat tároljuk a már felfedezett, legrövidebb távolsággal, mint kulcs értékkel.

Legrövidebb út - Dijkstra algoritmus

- Legyenek a nem KÉSZ csúcsok kékek, az elért csúcsok szürkék, a KÉSZ csúcsok pedig zöld színűek.
- A csúcsokra a címkén kívül, felírtuk az eddig talált legrövidebb út hosszát is (d tömbbeli értékeket).

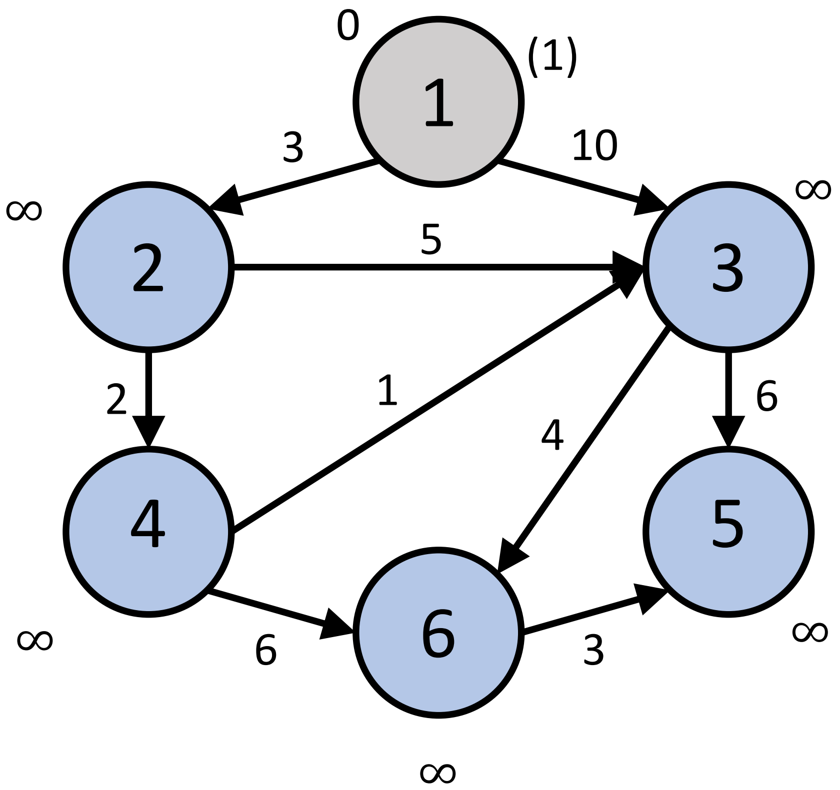
Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	∞	∞	∞	∞	∞
P	NIL	NIL	NIL	NIL	NIL	NIL

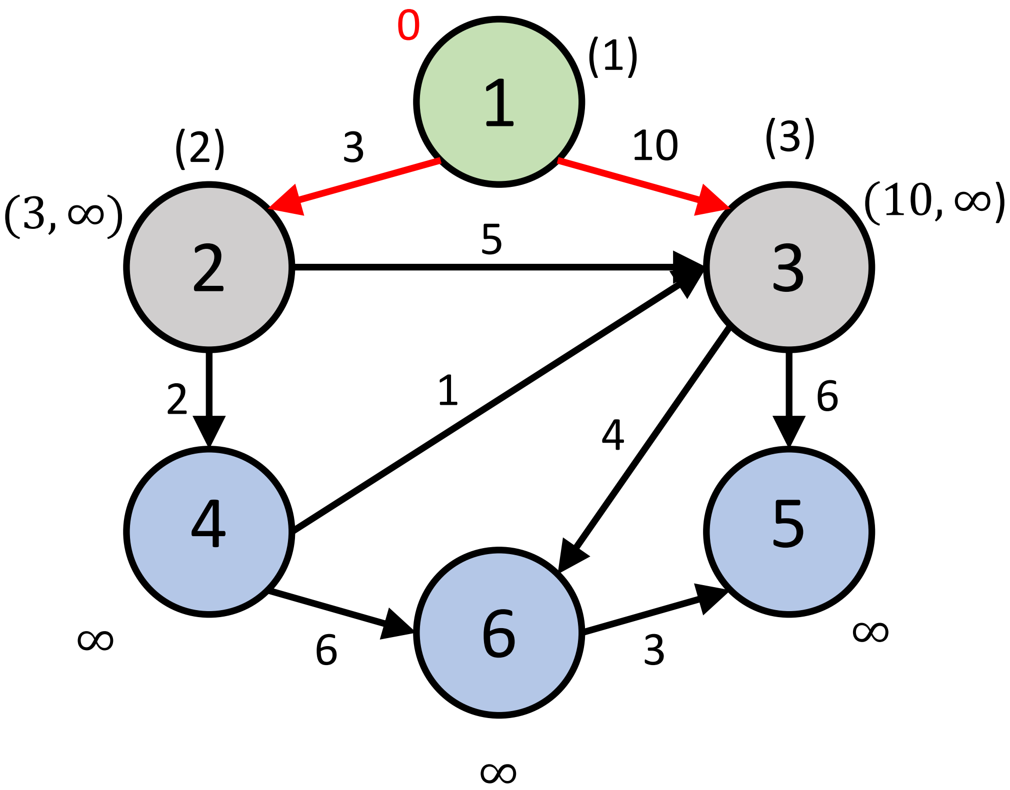
DEBRECENI EGYETEM

Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	∞	∞	∞	∞	∞
P	NIL	NIL	NIL	NIL	NIL	NIL

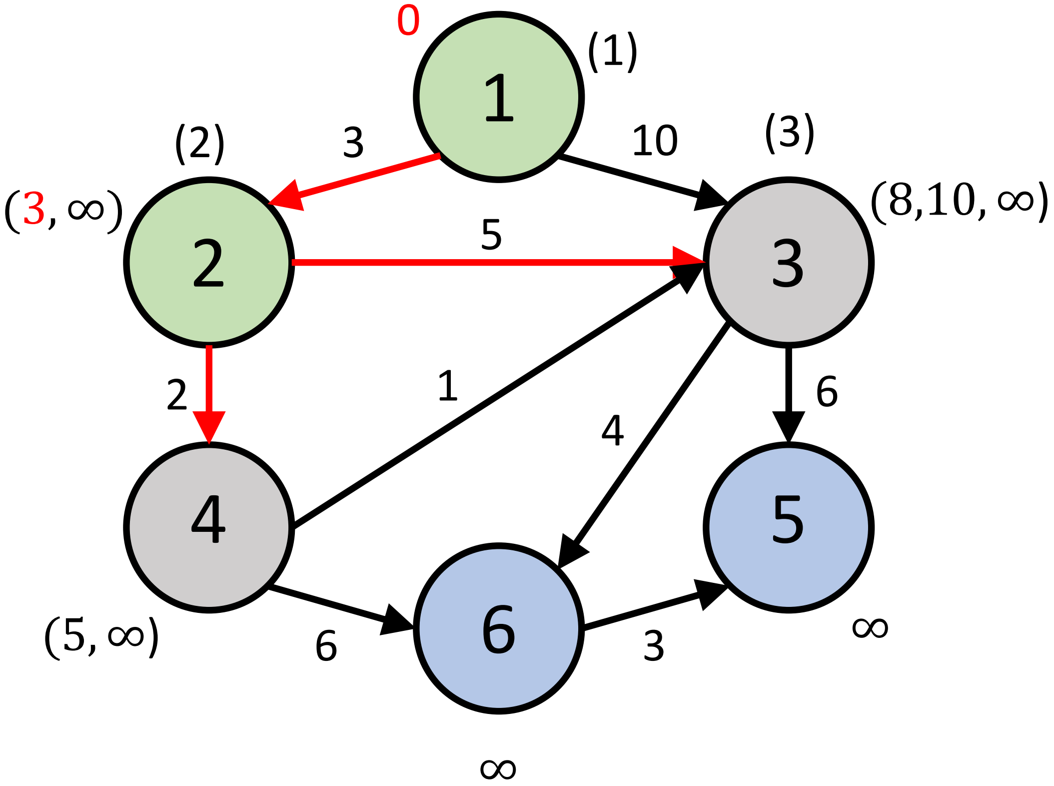
Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	3	10	∞	∞	∞
P	/	1	1	NIL	NIL	NIL

DEBRECENI EGYETEM

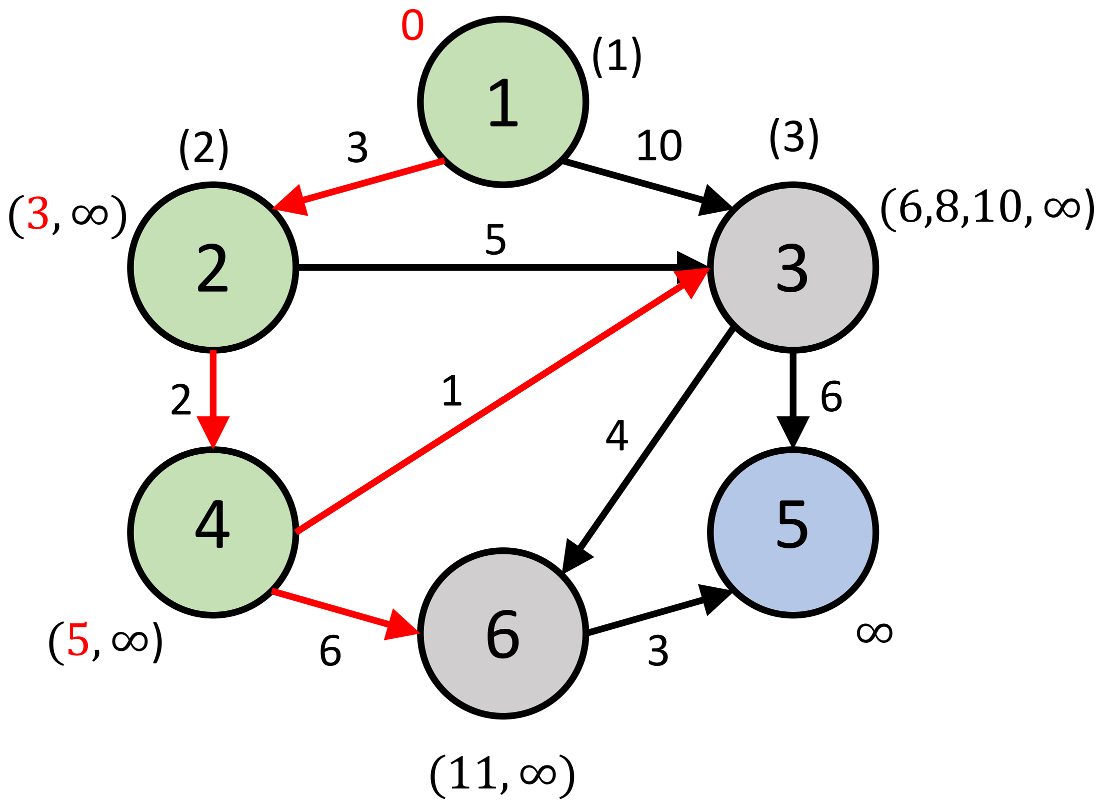
Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	3	8	5	∞	∞
P	/	1	2	2	NIL	NIL

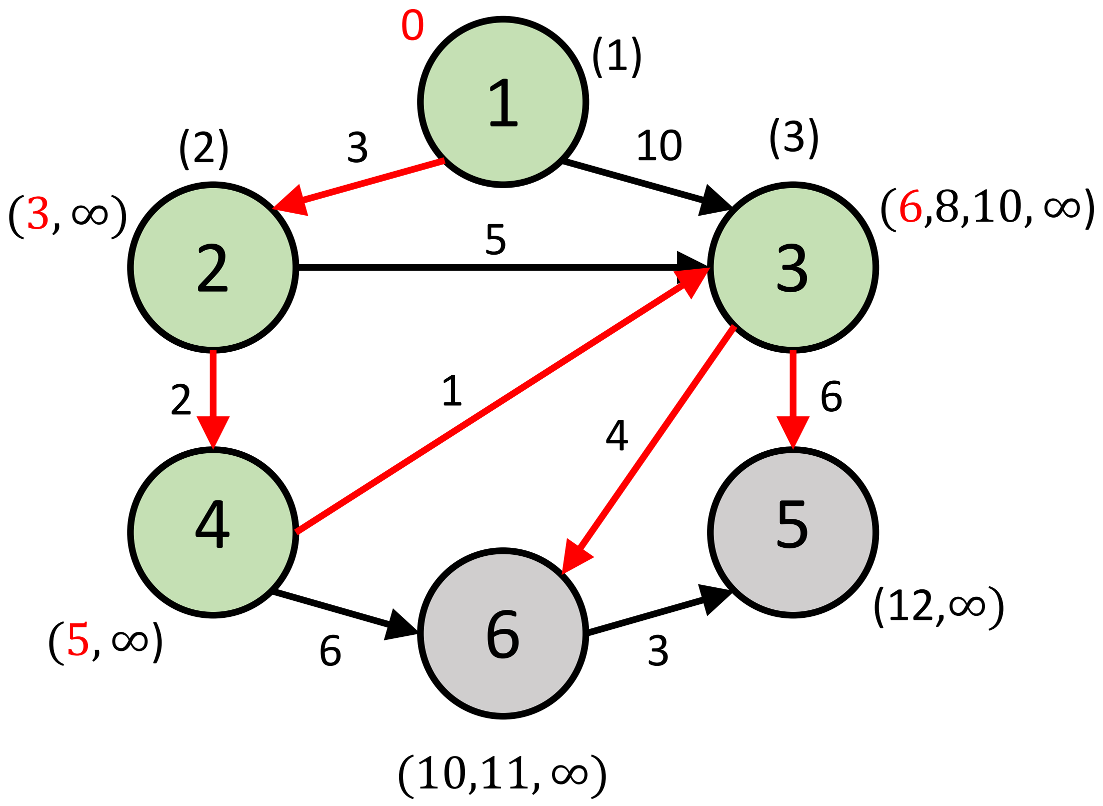
DEBRECENI EGYETEM

Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	3	6	5	∞	11
P	/	1	4	2	NIL	4

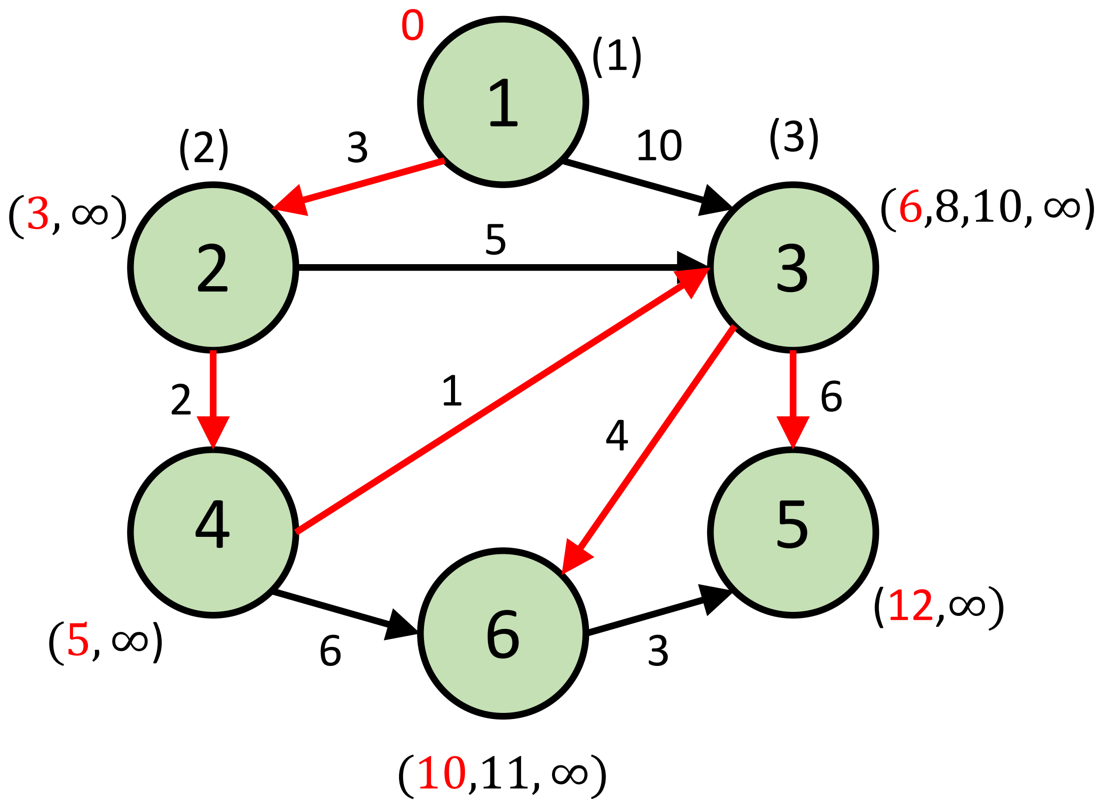
Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	3	6	5	12	10
P	/	1	4	2	3	3

DEBRECENI EGYETEM

Példa - Dijkstra algoritmus



	1	2	3	4	5	6
d	Ø	3	6	5	12	10
P	/	1	4	2	3	3

DEBRECENI EGYETEM

Példa – On-Line

Érdemes megnézni a következő videót is!

https://www.youtube.com/watch?v=pVfj6mxhdMw

DEBRECENI EGYETEM

Mohó algoritmus

- Egy nyugdíjas a postán szeretné felvenni a nyugdíját, aminek összege 79 500 Ft. Hogyan tudja a pénztáros a lehető legkevesebb bankjeggyel kifizetni azt?
- A feladat pontosítása érdekében tegyük fel, hogy
  - a rendelkezésre álló címletek: 500, 1000, 2000, 5000, 10000, 20000.
  - minden címletből tetszőleges mennyiség rendelkezésre áll.
  - a nyugdíjas nem rendelkezik pénzzel, így nem tud visszaadni.

DEBRECENI EGYETEM

Mohó algoritmus

- A mohó algoritmus lényege, – innen az elnevezés – hogy mindig a szóba jöhető címletek közül a legnagyobbat használjuk fel a hátralévő összeg kifizetésére.
  - 79 500 Ft=3x20 000 Ft+19 500 Ft
  - 19 500 Ft=1x10 000 Ft+9 500 Ft
  - 9 500 Ft =1x5 000 Ft+4 500 Ft
  - 4 500 Ft =2x2 000 Ft+500 Ft
  - 500 Ft =1x500 Ft
- A kifizetés legkevesebb 8 db bankjeggyel megoldható.
- Mohó algoritmus nem minden probléma esetén ad globális megoldást!

DEBRECENI EGYETEM

Algoritmusok „csoportosítása”

- Mohó algoritmusok (brute force)
- Oszd meg és uralkodj!
- Véletlent használó algoritmusok
- Prekondicionális algoritmusok
- Dinamikus programozás
- Lineáris programozás
- Közelítő algoritmusok

Dinamikus programozás

- A feladatot részfeladatokra való osztással oldja meg.
- Akkor alkalmazható, ha a részproblémák nem függetlenek, azaz közös részproblémáik vannak.
- Minden egyes részfeladatot (és annak részfeladatait is) pontosan egyszer oldja meg.
- A részfeladatok megoldásait tárolja, így elkerüli a felesleges számítást.

Köszönöm a figyelmet!