Informatika 1

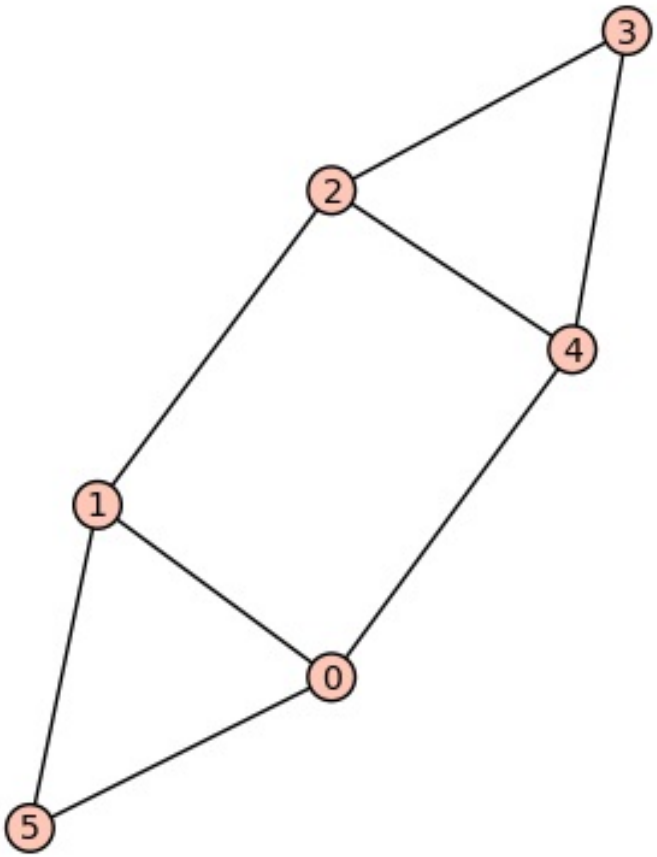
5. előadás

Kovács Kristóf

Budapesti Műszaki Egyetem

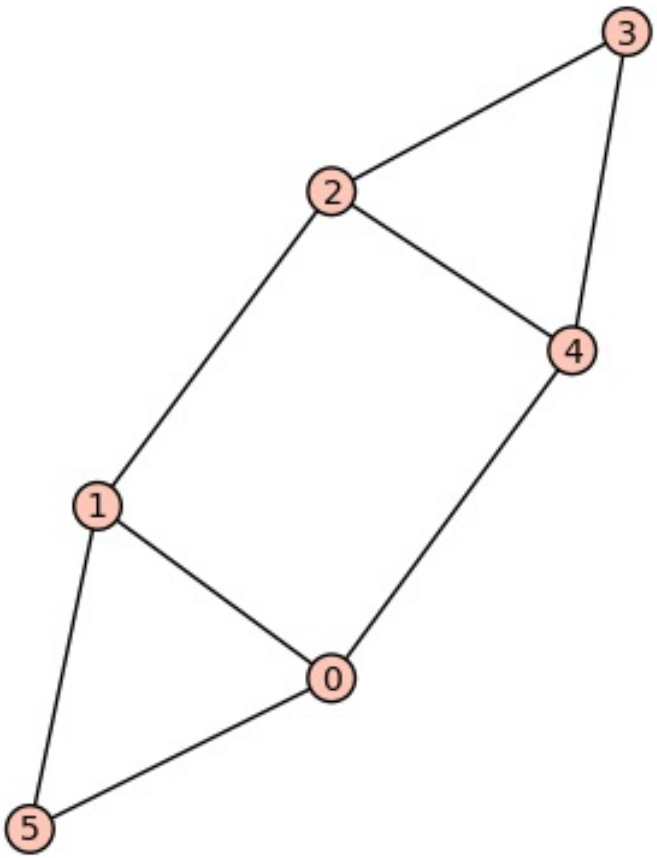
2013. október 9.

Gráfok reprezentálás



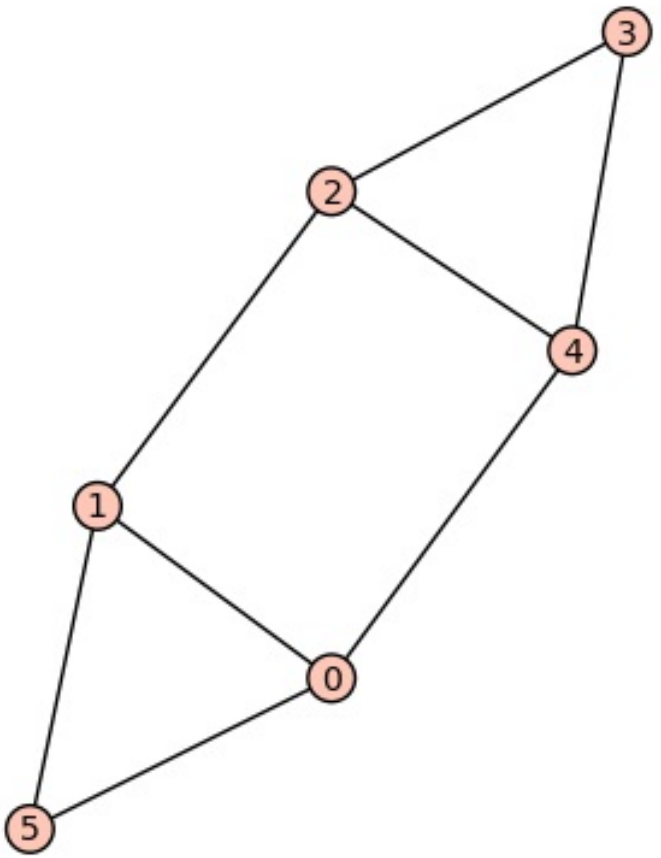
Gráfok reprezentálás

- Szomszédossági lista:
  $\{0 : [1, 4, 5], 1 : [2, 5],$
  $2 : [3, 4], 3 : [4], 4 : [], 5 : []\}$



Gráfok reprezentálás

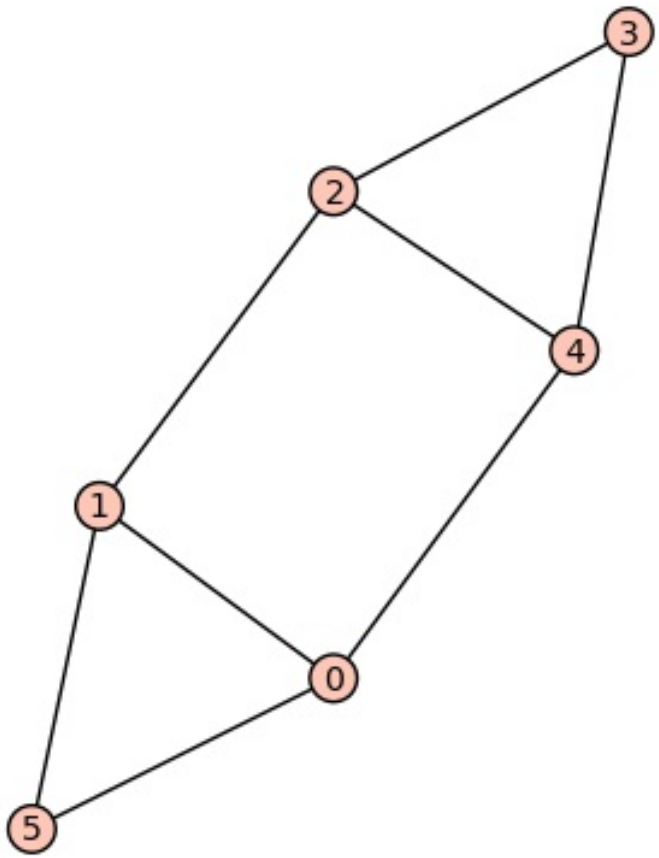
- Szomszédossági lista:
  $\{0 : [1, 4, 5], 1 : [2, 5],$
  $2 : [3, 4], 3 : [4], 4 : [], 5 : []\}$
- Éllista:
  $(0, 1), (0, 4), (0, 5), (1, 2),$
  $(1, 5), (2, 3), (2, 4), (3, 4)$



Gráfok reprezentálás

- Szomszédossági lista:
  $\{0 : [1, 4, 5], 1 : [2, 5],$
  $2 : [3, 4], 3 : [4], 4 : [], 5 : []\}$
- Éllista:
  $(0, 1), (0, 4), (0, 5), (1, 2),$
  $(1, 5), (2, 3), (2, 4), (3, 4)$
- Adjacencia mátrix:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Gráf sage-ben

- Legegyszerűbb a gráfokat szomszédossági listaval megadni:

```
G=Graph({0:[1,4,5],1:[2,5],2:[3,4],3:[4],4:[],5:[]})
```

Gráf sage-ben

- Legegyszerűbb a gráfokat szomszédossági listaval megadni:

```
G=Graph({0:[1,4,5],1:[2,5],2:[3,4],3:[4],4:[],5:[]})
```

- De mátrixszal is megy:

```
M = Matrix([[0,1,0,0,1,1],[1,0,1,0,0,1],[0,1,0,1,1,0],
            [0,0,1,0,1,0],[1,0,1,1,0,0],[1,1,0,0,0,0]])
G = Graph(M)
```

Gráf sage-ben

- Legegyszerűbb a gráfokat szomszédossági listaval megadni:

```
G=Graph({0:[1,4,5],1:[2,5],2:[3,4],3:[4],4:[],5:[]})
```

- De mátrixszal is megy:

```
M = Matrix([[0,1,0,0,1,1],[1,0,1,0,0,1],[0,1,0,1,1,0],
            [0,0,1,0,1,0],[1,0,1,1,0,0],[1,1,0,0,0,0]])
G = Graph(M)
```

- Amint látjátok a mátrixok listába ágyazott listák.
- A *DiGraph* paranccsal tudunk irányított gráfokat definiálni. Ekkor a szomszédossági listában a kulcs lesz ahonnan kiindulnak az élek és a listában szereplő csúcsokba mennek.

Gráfok hasznos metódusai

- *G.show()* – kirajzolja a gráfot
- *G.degree()* – megadja egy listában a csúcsok fokát
- *G.degree(2)* – megadja a 2-es csúcs fokát
- *G.edges()* – visszaadja a gráf éllistáját
- *G.neighbors(5)* – visszaadja az 5-ös csúcs szomszédait
- *G.distance(2,5)* – a 2-es és 5-ös csúcs közti út hossza

Szimbolikus számolás sage-ben

- Sage-el könnyedén megoldhatunk egyenleteket, egyenletrendszereket, kiszámolhatunk ocsmány deriváltakat, integrálokat.

Szimbolikus számolás sage-ben

- Sage-el könnyedén megoldhatunk egyenleteket, egyenletrendszereket, kiszámolhatunk ocsmány deriváltakat, integrálokat.
- Az első dolgunk, hogy deklarálunk szimbolikus változókat:

```
a = var('a')
b = var('b')
```

vagy összevonva:

```
a,b = var('a,b')
```

Szimbolikus számolás sage-ben

- Sage-el könnyedén megoldhatunk egyenleteket, egyenletrendszereket, kiszámolhatunk ocsmány deriváltakat, integrálokat.
- Az első dolgunk, hogy deklarálunk szimbolikus változókat:

```
a = var('a')
b = var('b')
```

vagy összevonva:

```
a,b = var('a,b')
```

- Ezután használhatjuk az a és b változót szimbolikus függvényekben, metódusokban, pl:

```
expand((a + b)^2)
```

Eredménye:

```
a^2 + 2*a*b + b^2
```

Hasznos szimbolikus függvények

- *solve(egyenlet, vált)* – megoldja szimbolikusan az egyenletet

```
x = var('x')
solve(x^2 + 3*x +2, x)
```

Eredmény:

```
[x == -2, x == -1]
```

Hasznos szimbolikus függvények

- *solve(egyenlet, vált)* – megoldja szimbolikusan az egyenletet

```
x = var('x')
solve(x^2 + 3*x +2, x)
```

Eredmény:

```
[x == -2, x == -1]
```

- *find_root(egyenlet, l, u)* – numerikusan keres egy gyököt

```
find_root(sin(x) == cos(x), 0, pi/2)
```

Eredmény:

```
0.7853981633974484
```

Még példák

- *solve([egyenletek], változók)* – egyenletrendszert old meg

  ```
  solve([x + y == 6, x - y == 4], x, y)
  ```

  Eredmény:

  ```
  [[x == 5, y == 1]]
  ```

Még példák

- *solve([egyenletek], változók)* – egyenletrendszert old meg

  ```
  solve([x + y == 6, x - y == 4], x, y)
  ```

  Eredmény:

  ```
  [[x == 5, y == 1]]
  ```

- *roots(egyenlet, változó)* – multiplicitással megadja a gyököket

  ```
  roots(x**3 - x**2 - x + 1 == 0, x)
  ```

  Eredmény:

  ```
  [(-1, 1),(1, 2)]
  ```