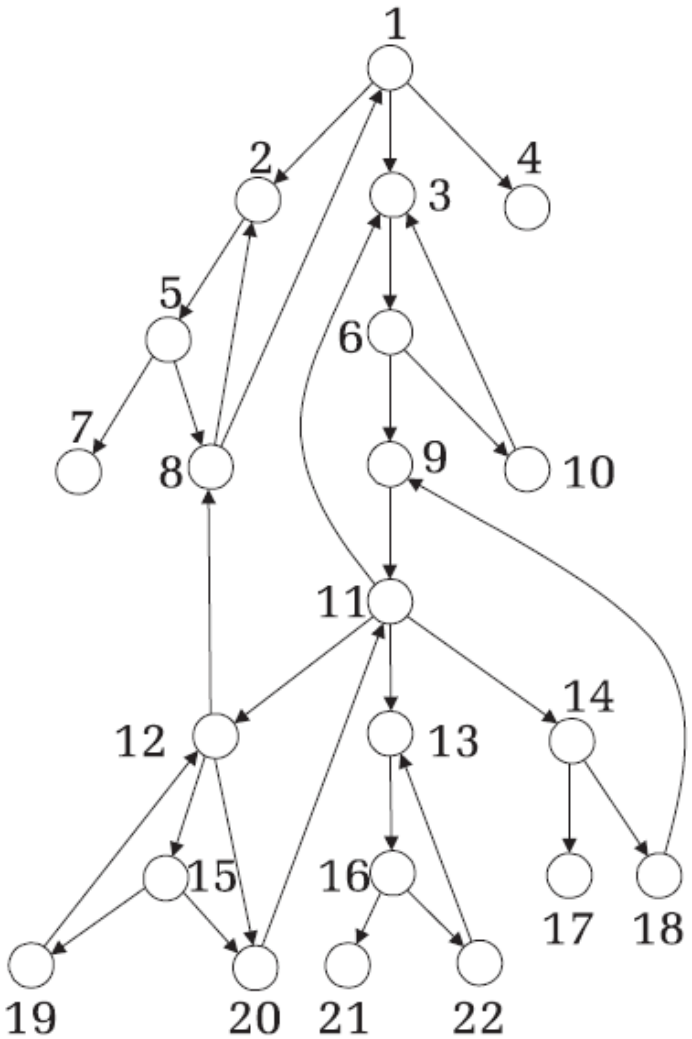
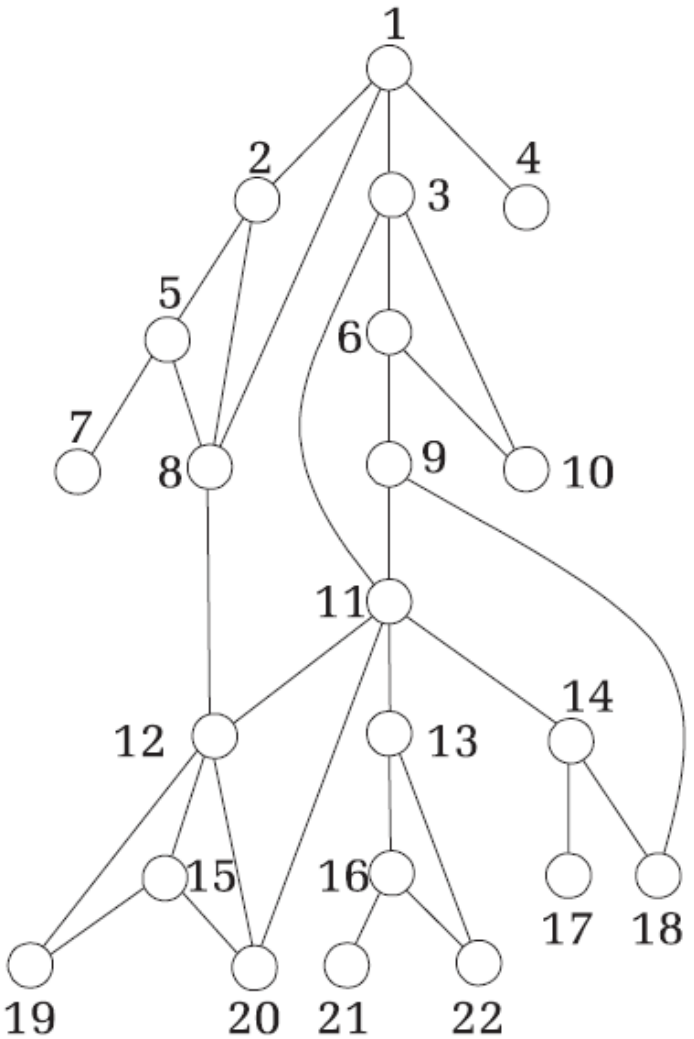
Prog2
3. előadás

2020-21

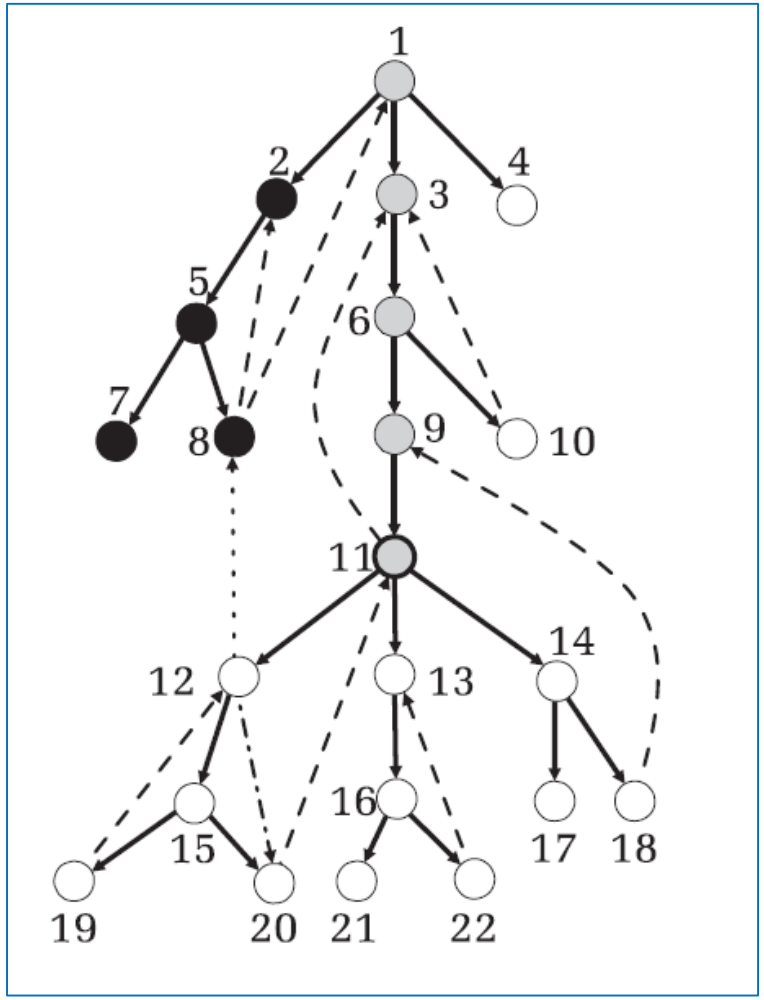
DFS (mélységi bejárás; irányított vs. irányítatlan)

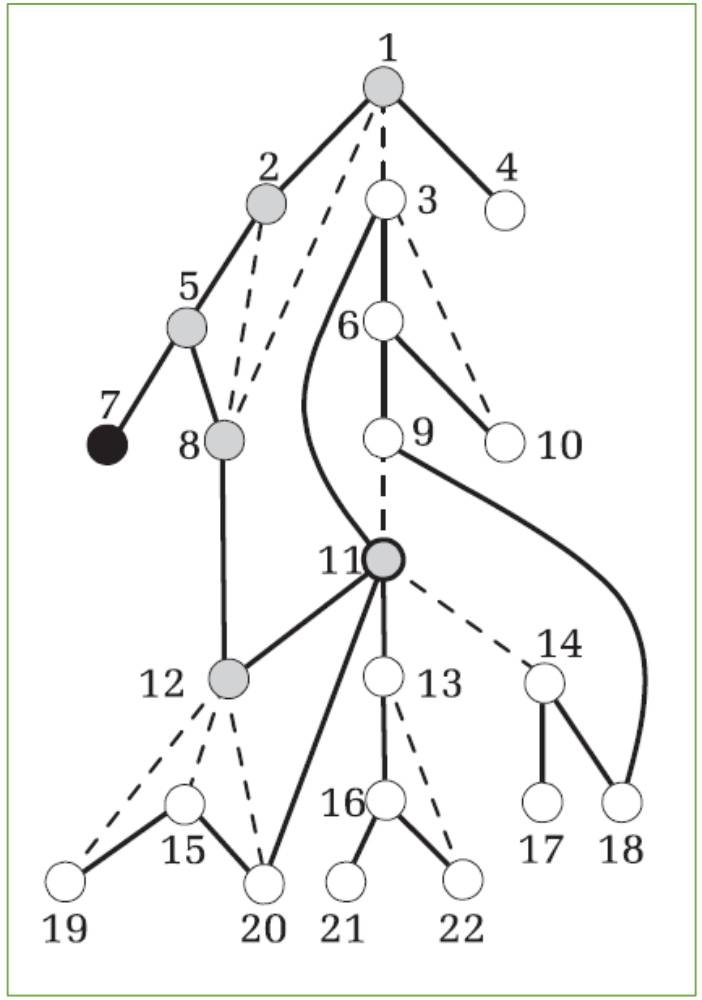




DFS

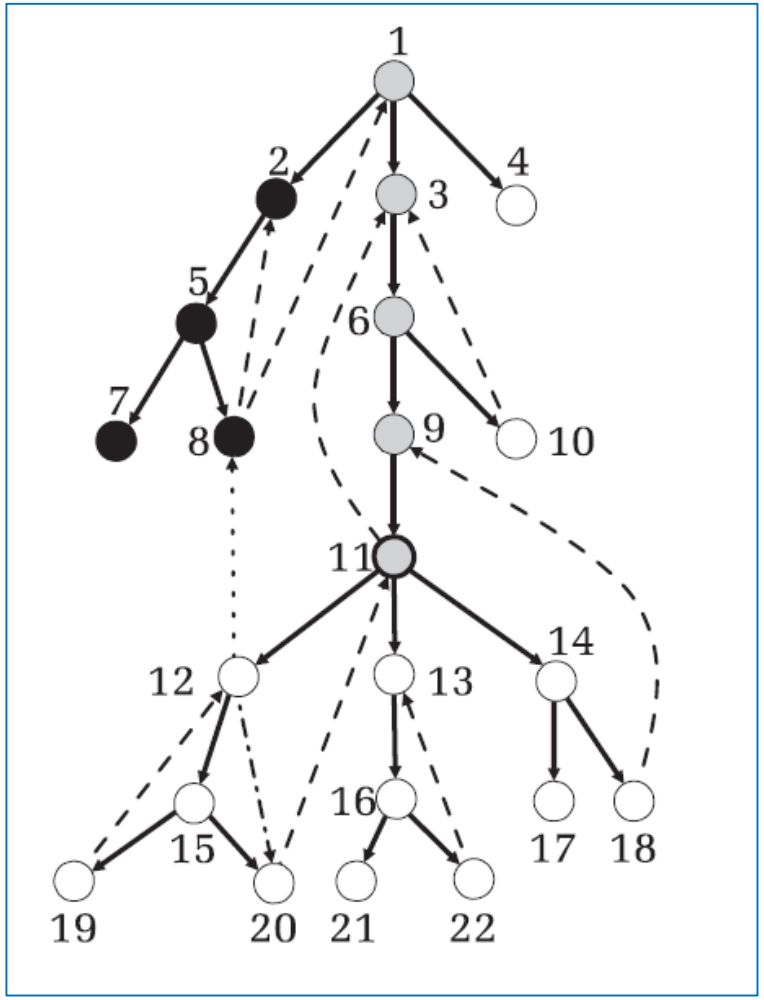
DFS-sorrend: **1, 2, 5, 7, 8, 12, 11, 3, 6, 9, 18, 14, 17, 10, 13, 16, 21, 22, 20, 15, 19, 4**

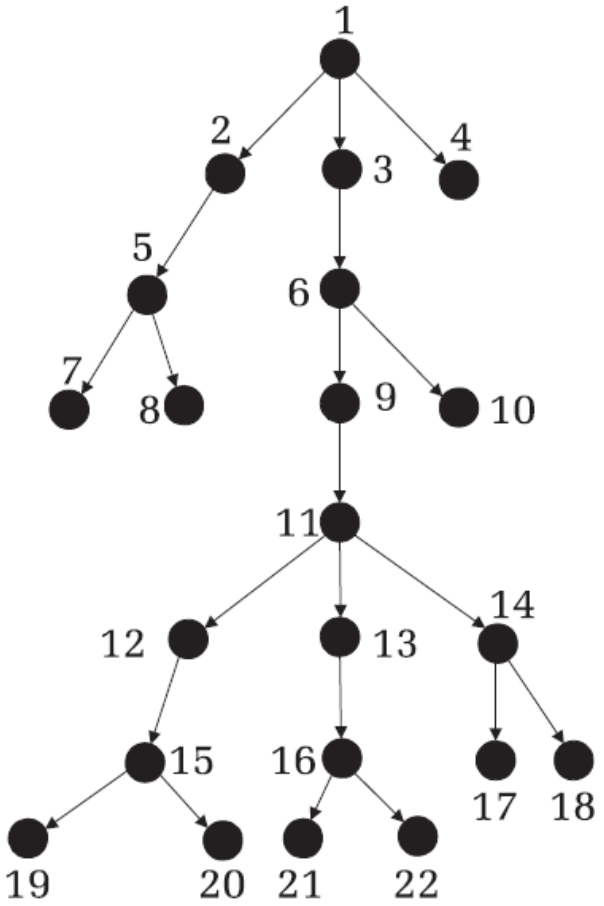




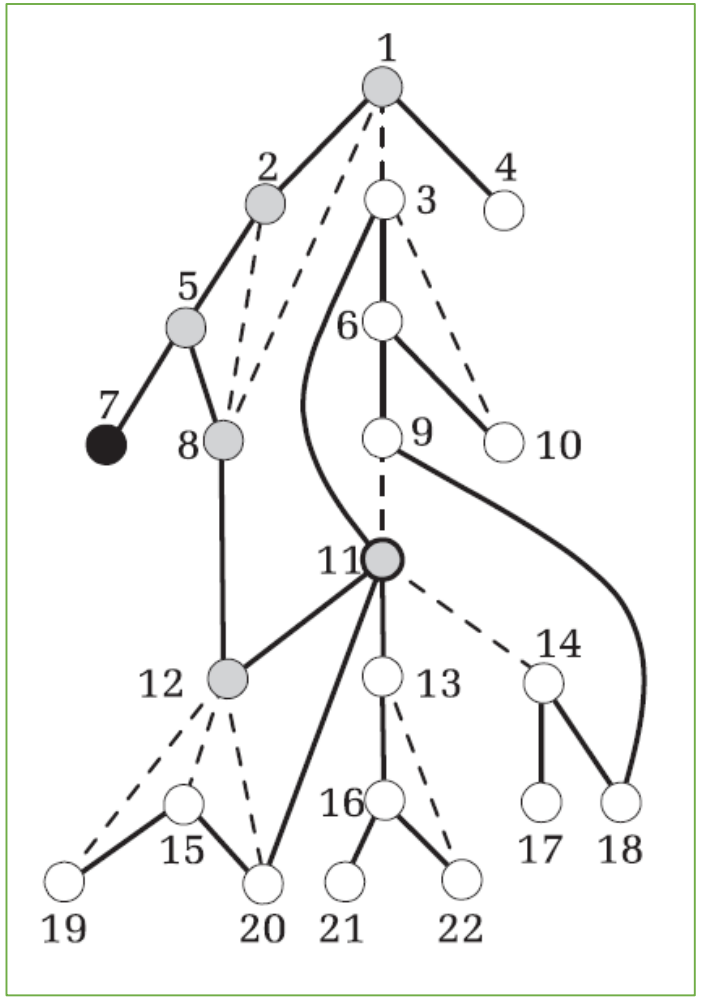
DFS-sorrend: **1, 2, 5, 7, 8, 3, 6, 9, 11, 12, 15, 19, 20, 13, 16, 21, 22, 14, 17, 18, 10, 4**

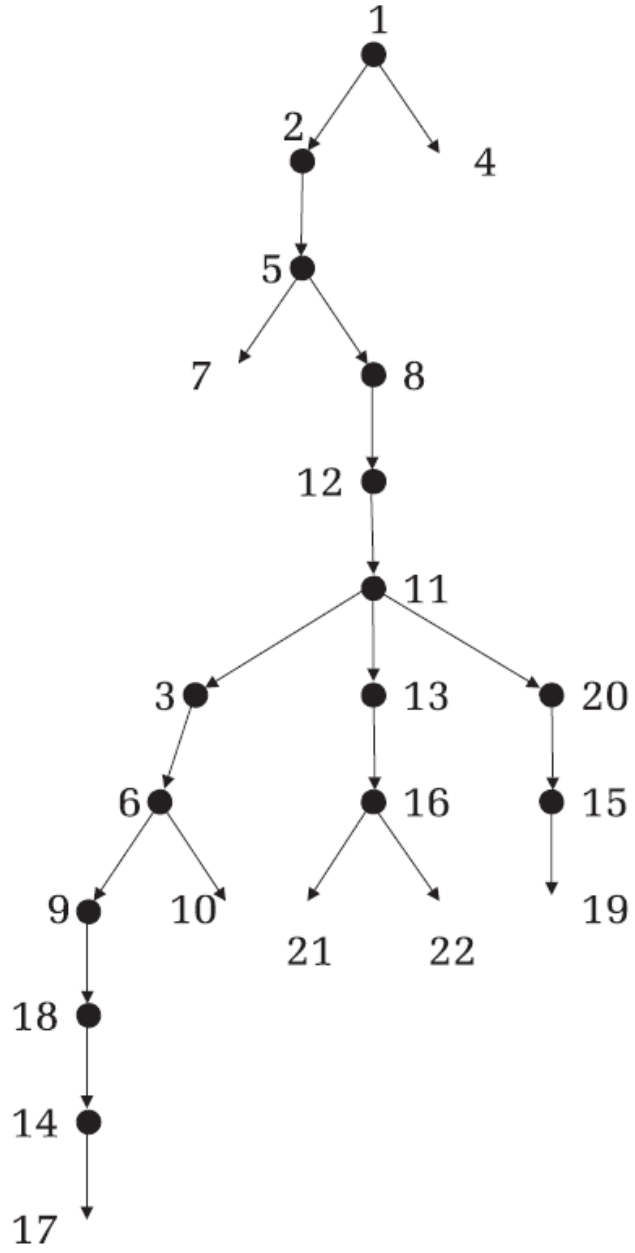
DFS (mélységi fa)





DFS (mélységi fa)





	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Apa	0	1	11	1	2	3	5	5	6	6	12	8	11	18	20	13	14	9	15	11	16	16

DFS (a kód)

```
eljárás MÉLYSÉGI_BEJÁRÁS(G)
    minden u ∈ V(G) végezd
        szín[u] ← FEHÉR
        apa[u] ← 0
    vége minden
    minden u ∈ V(G) végezd
        ha szín[u] = FEHÉR akkor
            MÉLYSÉGI_MENET(G,u)
        vége ha
    vége minden
vége MÉLYSÉGI_BEJÁRÁS
```

```
eljárás MÉLYSÉGI_MENET(G,u)
    kiír: u
    szín[u] ← SZÜRKE
    minden v ∈ Szomszéd(u) végezd
        ha szín[v] = FEHÉR akkor
            apa[v] ← u
            MÉLYSÉGI_MENET(G,v)
        vége ha
    vége minden
    szín[u] ← FEKETE
vége MÉLYSÉGI_MENET
```

Körmentesség

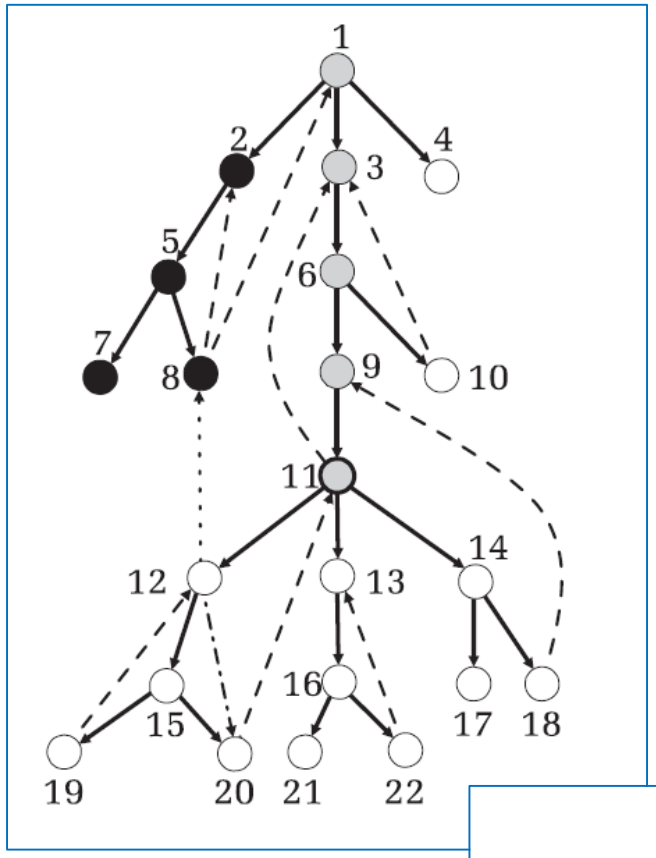
Irányított gráfok esetén:

```
...
különben
    ha szín[v] = SZÜRKE akkor
        körmentes ← HAMIS
    vége ha
...
```

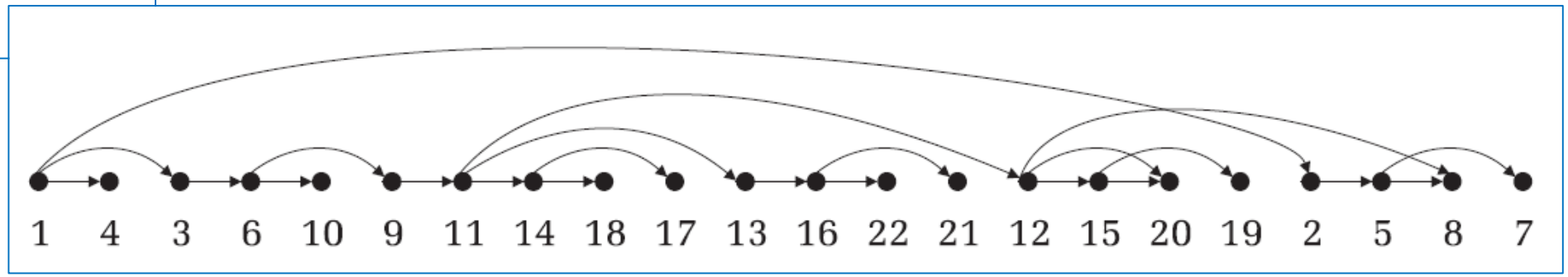
Irányítatlan gráfok esetén:

```
...
különben
    ha (szín[v] = SZÜRKE) ÉS (apa[v] ≠ u) akkor
        körmentes ← HAMIS
    vége ha
...
```

Topologikus rendezés




```
eljárás MÉLYSÉGI_MENET(G,u)
    szín[u] ← SZÜRKE
    minden v ∈ Szomszéd(u) végezd
        ha szín[v] = FEHÉR akkor
            apa[v] ← u
            MÉLYSÉGI_MENET(G,v)
        vége ha
    vége minden
    szín[u] ← FEKETE
    kiír: u
vége MÉLYSÉGI_MENET
```

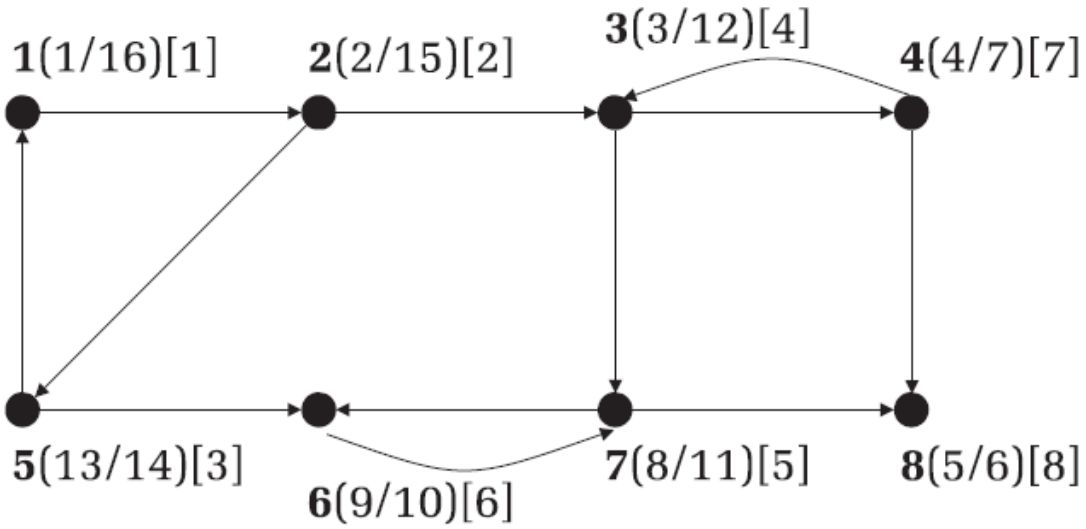


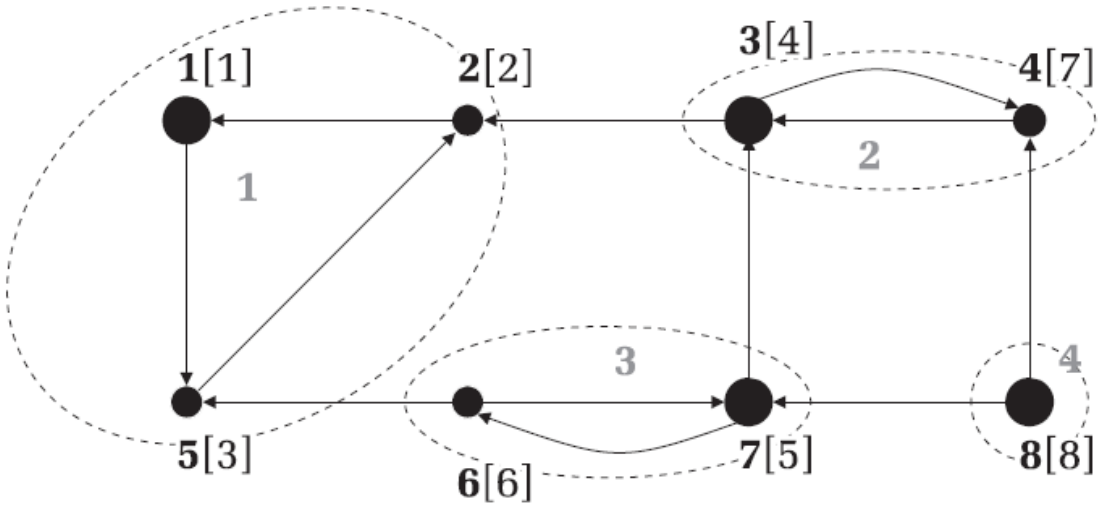
Irányítatlan gráfok **összefüggő** komponensei

```
eljárás MÉLYSÉGI_BEJÁRÁS(G)
    minden u ∈ V(G) végezd
        szín[u] ← FEHÉR
        apa[u] ← 0
    vége minden
    minden u ∈ V(G) végezd
        ha szín[u] = FEHÉR akkor
            MÉLYSÉGI_MENET(G,u)
        vége ha
    vége minden
vége MÉLYSÉGI_BEJÁRÁS
```

Minden mélységi menet letapogat egy komponenst

Irányított gráfok **erősen összefüggő** komponensei





1. DFS ki-szomszédsági listán
   - Felállítjuk a fordított feketévé válás sorrendjét (**ff**)
2. DFS be-szomszédsági listán úgy, hogy a pontokat a fordított feketévé válási sorrendjükben tekintjük
   - A letapogatott komponensek a gráf erősen összefüggő komponensei lesznek

Erősen összefüggő komponensek

```
eljárás MÉLYSÉGI_BEJÁRÁS(kiG)
    …
    minden u = 1,n végezd
        ha szín[u] = FEHÉR akkor
            MÉLYSÉGI_MENET(kiG,u)
        vége ha
    vége minden
vége MÉLYSÉGI_BEJÁRÁS
```

```
eljárás MÉLYSÉGI_BEJÁRÁS(beG)
    …
    minden u = 1,n végezd
        ha szín[ff[u]] = FEHÉR akkor
            MÉLYSÉGI_MENET(beG,ff[u])
            Kiír: '\n'
        vége ha
    vége minden
vége MÉLYSÉGI_BEJÁRÁS
```

```
eljárás MÉLYSÉGI_MENET(kiG,u)
    minden v ∈ kiSzomszéd(u) végezd
        ha szín[v] = FEHÉR akkor
            MÉLYSÉGI_MENET(kiG, v)
        vége ha
    vége minden
    ff[--k] ← u
vége MÉLYSÉGI_MENET
```

```
eljárás MÉLYSÉGI_MENET(beG,u)
    minden v ∈ beSzomszéd(u) végezd
        ha szín[v] = FEHÉR akkor
            MÉLYSÉGI_MENET(beG, v)
        vége ha
    vége minden
vége MÉLYSÉGI_MENET
```