Přednáška 4

Prohledávání grafu do hloubky

V této části probereme témata:

- prohledání grafu do hloubky (DFS), strom prohledání do hloubky (DF-strom), časové značky uzlů, časová složitost prohledání do hloubky
- algoritmus topologického uspořádání uzlů
- algoritmus určení silných komponent orientovaného grafu

Skripta odstavec 4.2, str. 79 - 90

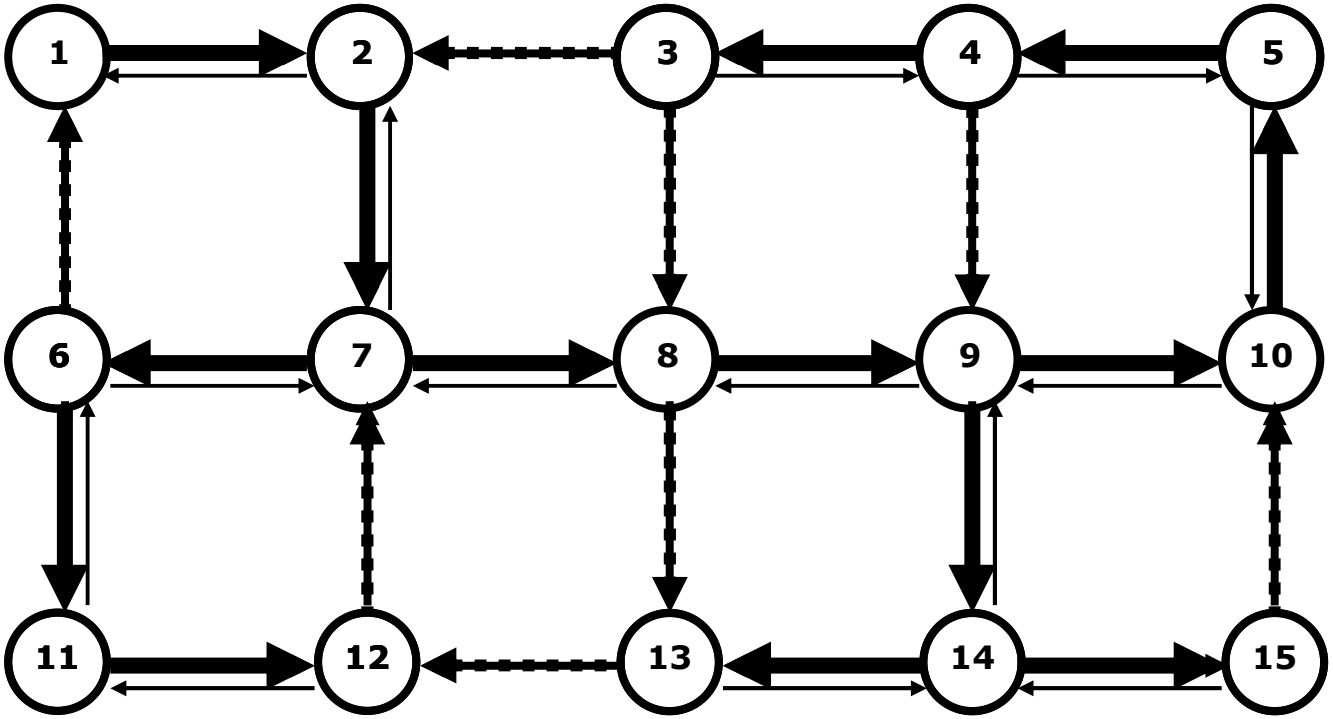
Prohledávání grafu do hloubky

Základní princip DFS :

postupuje se stále dál od počátečního uzlu dosud neprozkoumaným směrem. Když už to dál nejde, vrátíme se a postupujeme zase co nejdál.

Pro jednoduchost předpokládáme, že

sousedi jsou řazeni v pořadí rostoucích pořadových čísel uzlů


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Jak budeme DFS implementovat ?

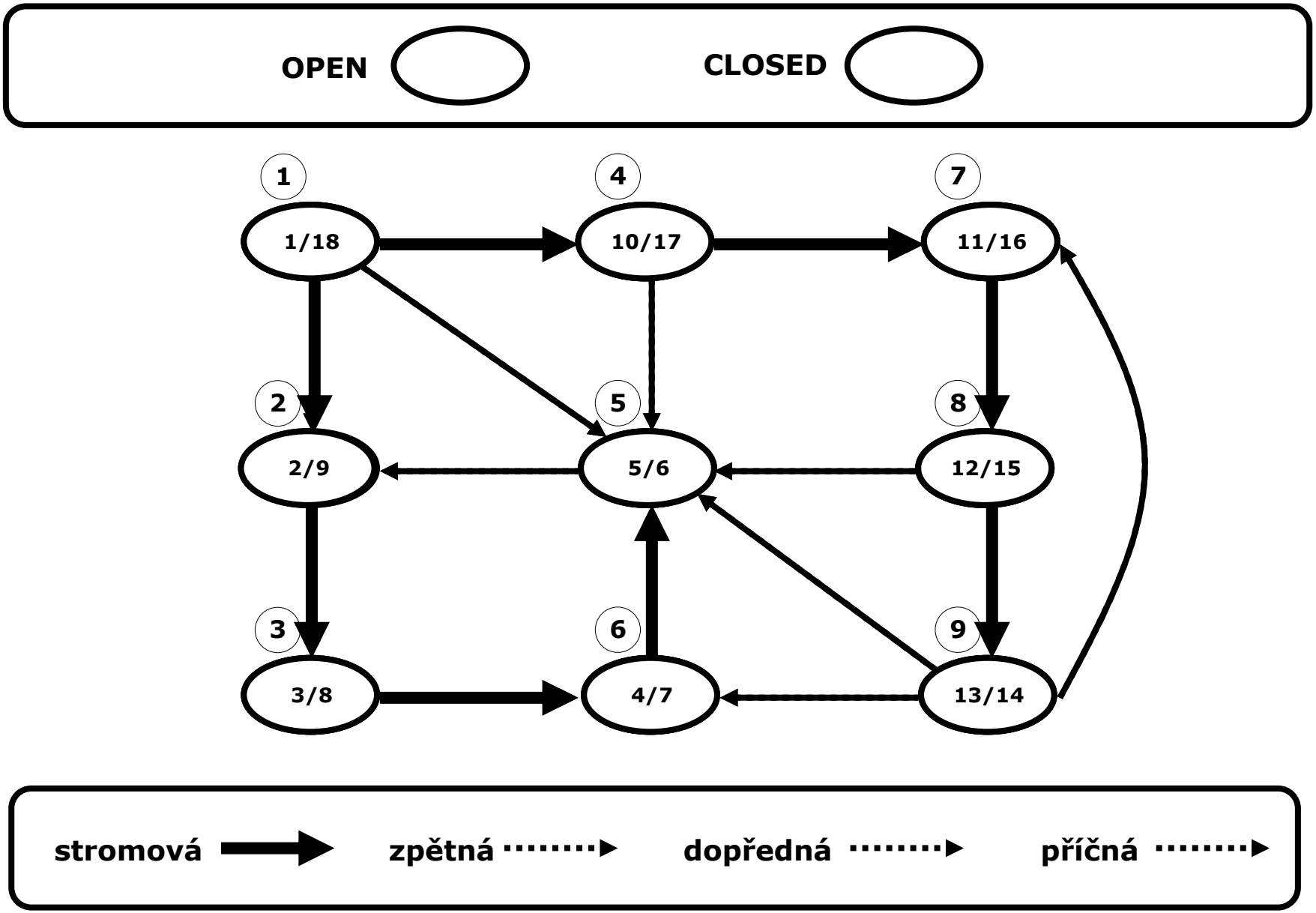
DFS - Depth-First Search

Výsledkem bude DF-strom (nebo les)

Uzly jsou opět FRESH, OPEN nebo CLOSED, ale mají **časové značky** s hodnotami 1 … 2*|U|

- **d[u]** se uzlu přidělí se při otevření
- **f[u]** se uzlu přidělí se při uzavření (takže d[u]<f[u])

```
   void DFS (Graph G) {    // pseudokód
1     for (Node u in U(G))
2       { stav[u] = FRESH;  p[u] = null; }
3     i = 0;
4     for (Node u in U(G))
5       if (stav[u] == FRESH) DFS-Projdi(u);
6   }
   void DFS-Projdi(Node u) {
1     stav[u] = OPEN; d[u] = ++i;
2     for (Node v in Adj[u])
3       if (stav[v] == FRESH)  {
4         p[v] = u;  DFS-Projdi(v); }
5     stav[u] = CLOSED; f[u] = ++i;
6   }
```


OPEN
CLOSED
1
4
7
1/18
10/17
11/16
2
5
8
2/9
5/6
12/15
3
6
9
3/8
4/7
13/14
stromová
zpětná
dopředná
příčná

Jak složitý je algoritmus DFS ?

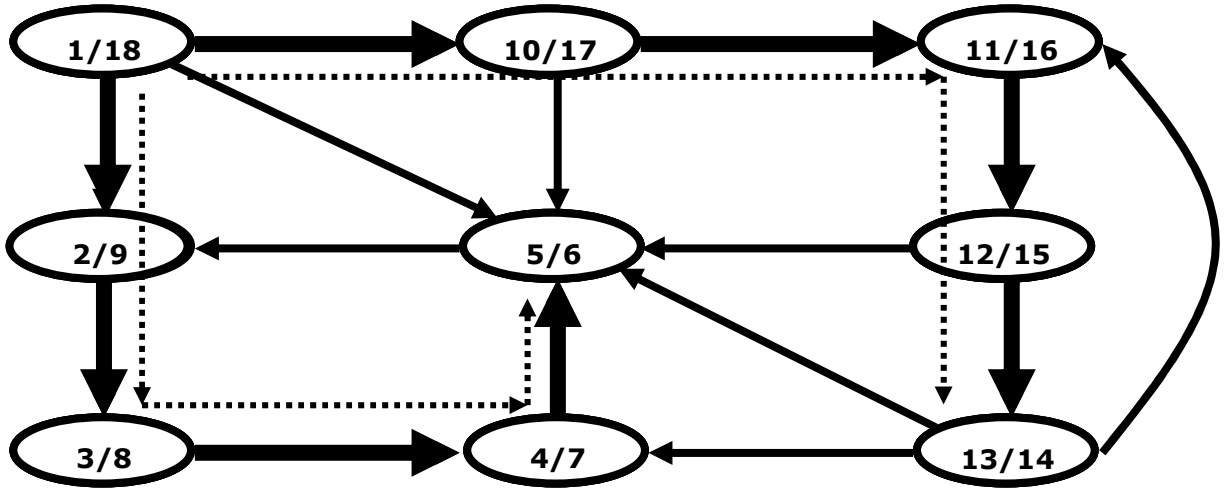
- v DFS se cykly na řádcích 1 - 2 a 4 – 5 provedou |U|-krát
- DFS-Projdi se volá |U|-krát
- cykl v DFS-Projdi na ř. 2-4 se provádí |Adj[u]|-krát $\Rightarrow$ dohromady O(|H|)

$$\Rightarrow \mathbf{O(|U| + |H|)}$$

K čemu je dobré značkování uzlů ?

Obecně platí: $(d[u],f[u]) \cap (d[v],f[v]) = \emptyset$ nebo

$(d[u],f[u]) \subset (d[v],f[v])$ nebo $(d[v],f[v]) \subset (d[u],f[u])$

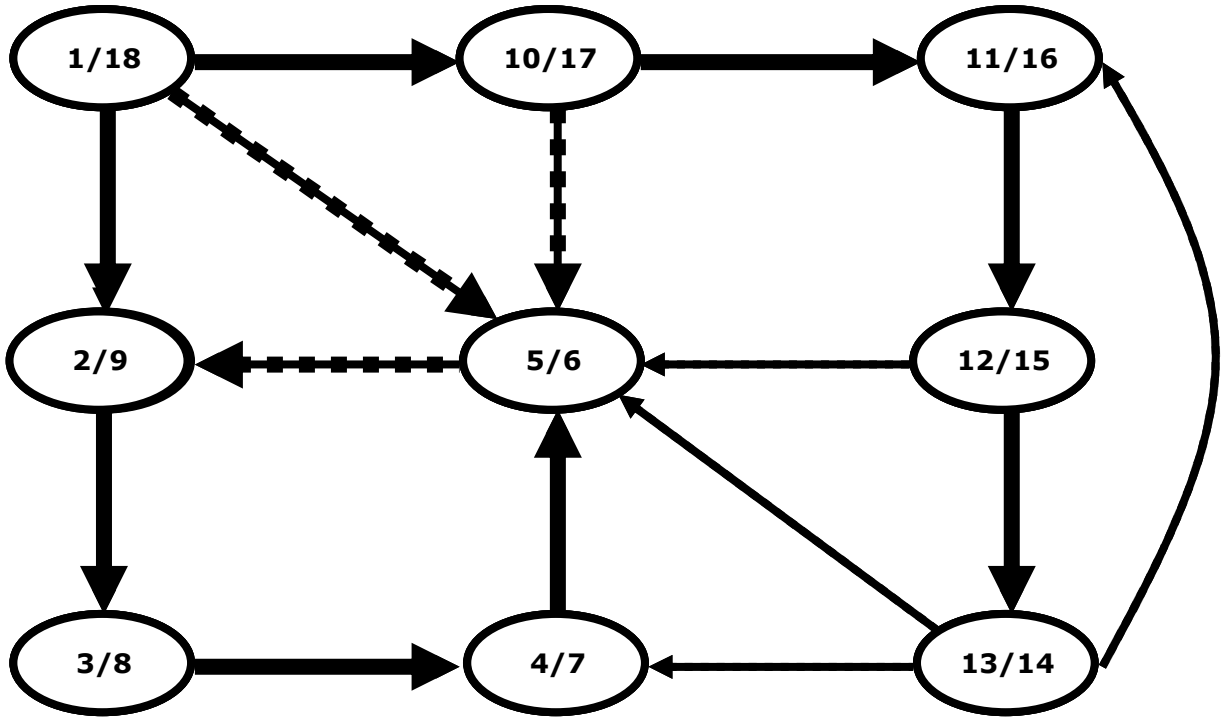


Sledujme stromové hrany:

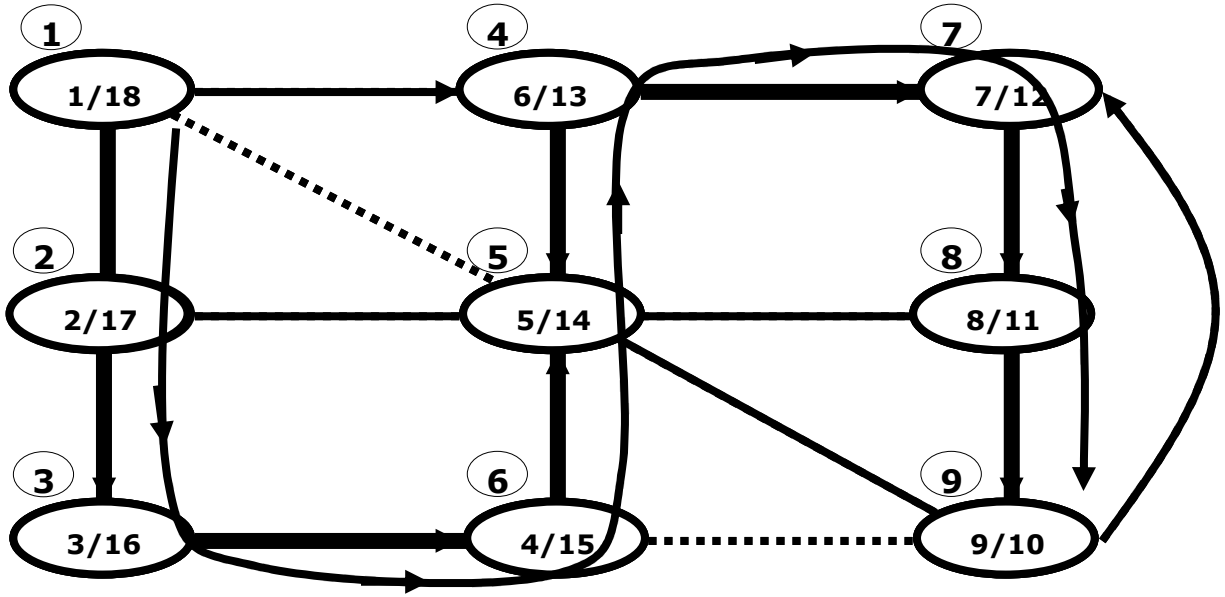
(1 .. 18) ⊃ (2 .. 9) ⊃ (3 .. 8) ⊃ (4 .. 7) ⊃ (5 .. 6)

(1 .. 18) ⊃ (10 .. 17) ⊃ (11 .. 16) ⊃ (12 .. 15) ⊃ (13 .. 14)

- **zpětná hrana** (u,v) vede k OPEN uzlu v
- **dopředná a příčná hrana** (u,v) vede ke CLOSED uzlu
  - **dopředná má d[u]<d[v]**
  - **příčná má d[u]>d[v]**



Jak se liší DFS u neorientovaného grafu ?



Jsou zde pouze stromové a zpětné hrany !

Topologické uspořádání uzlů pomocí DFS

Top-Sort-1 (G)

1) **S:=∅**
2) **prováděj DFS(G) a v okamžiku f[u] ulož uzel u na začátek seznamu S**
3) **S obsahuje uzly v topologickém uspořádání**

V: G je acyklický $\Leftrightarrow$ DFS(G) neobjeví zpětnou hranu

Top-Sort-2 (G)

eliminací kořenů - viz dříve

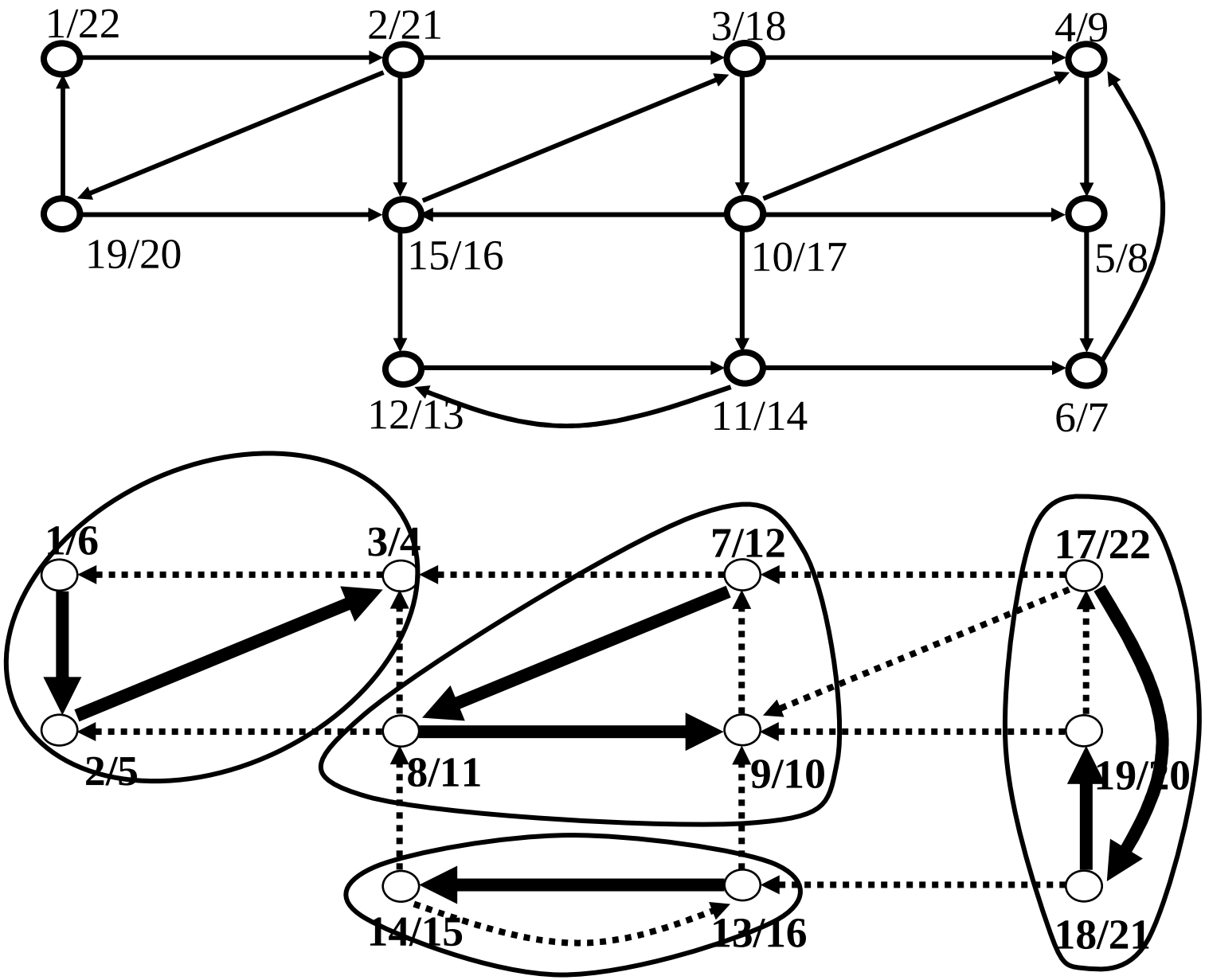
$\delta[u] = \delta^-_G[u]$

M - množina kořenů (fronta)

Silné komponenty pomocí DFS

S-COMP (G)

1) **DFS(G) určí d[u] & f[u] pro všechny u ∈ U**
2) **vytvoří se G^- (opačně orientovaný graf)**
3) **provede se DFS(G^-) s tím, že uzly v hlavním cyklu se berou v klesajícím pořadí f[u]**
4) **stromy DF-lesa určují silné komponenty**


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

Pozvánka na přednášku

Ve středu 23.3. 2010 od 12:45 v E-301 (K9)

přednáší Ing. Petr Budiš, CSc.,

ředitel První certifikační autority a.s. (ICA)

Kontrolní otázky

4.1 Vysvětlete na příkladu, že je možné, aby uzel u orientovaného grafu skončil při prohledání do hloubky jako jediný uzel nějakého dílčího DF-stromu, přestože do u vcházejí i z něho vycházejí hrany.

4.2 Navrhněte obecný postup orientace hran neorientovaného grafu, jehož výsledkem bude acyklický orientovaný graf.

4.3 Upravte algoritmus procházení neorientovaného (resp. orientovaného) grafu do hloubky tak, aby generoval všechny cesty (resp. orientované cesty) vycházející ze zadaného počátečního uzlu s.

4.4 Navrhněte algoritmus časové složitosti O(|U|+|H|), který pro zadaný acyklický orientovaný graf určí počet (nikoliv nutně strukturu) všech orientovaných cest (délky alespoň 1) v tomto grafu.

4.5 Navrhněte algoritmus časové složitosti O(|U|), který zjistí, zda je zadaný neorientovaný graf stromem.

4.6 Při procházení orientovaného grafu G do hloubky byla zjištěna existence hran všech čtyř typů (tj. stromové, zpětné, dopředné i příčné). Pro které z následujících vlastností lze z této skutečnosti odvodit nějaký závěr a jaký? a) souvislost b) silná souvislost c) acykličnost

4.7 Jaký je hlavní rozdíl algoritmů BFS a DFS pro obecné grafy a algoritmů pro systematický průchod kořenových stromů?