

# Projektovanje algoritama

L10. Grafovi. Pretraga grafova. BFS. DFS

# Današnje teme

- Struktura podataka – graf
- Implementacija grafova
- Pretraga grafova. BFS. DFS
- Računanje udaljenosti dve tačke u grafu
  
- **Lektira:** 22. Elementary Graph Algorithms
  - 22.1, 22.2, 22.3 [str. 587-612]

# Graf

$$G = (V, E)$$

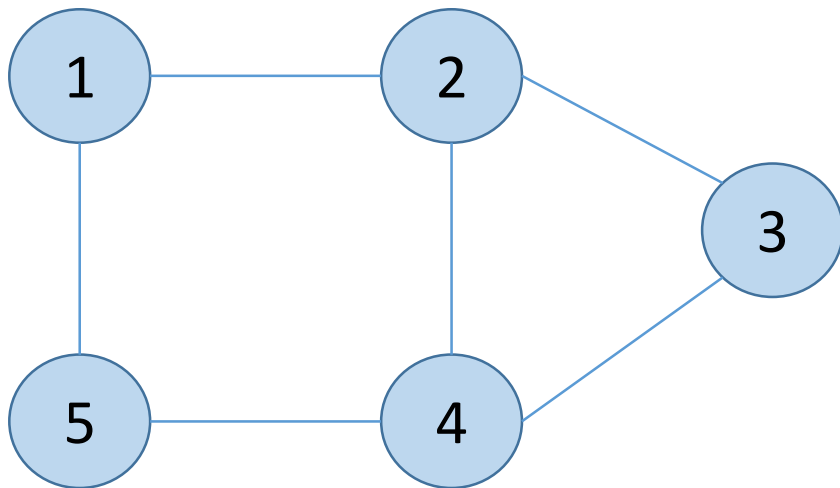
$V$  – skup čvorova grafa

$E$  – skup ivica grafa (ivica povezuje čvorove)

Tipovi grafova:

1. Usmereni graf (*directed graph*)
2. Neusmereni graf (*undirected graph*)

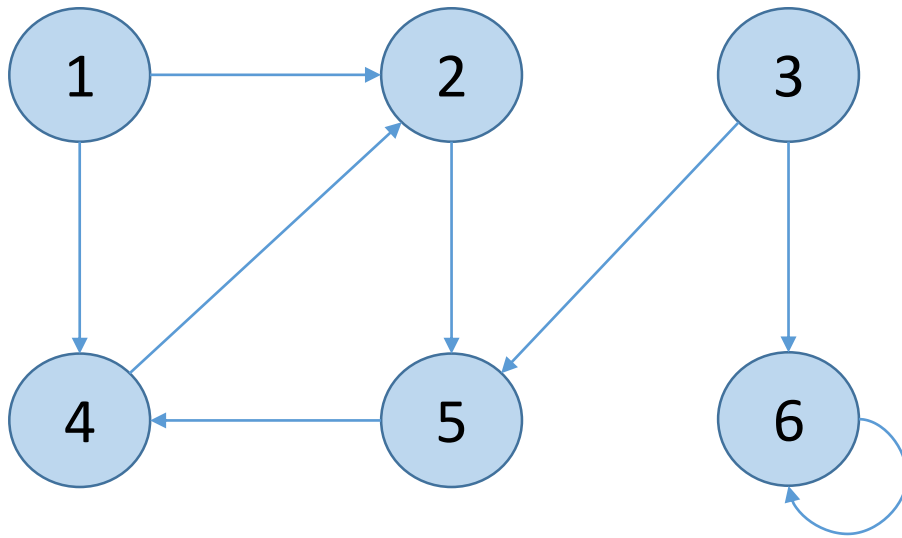
# Neusmereni graf



Ivica  $(x, y)$  znači da je:

- $x$  povezan sa  $y$
- $y$  povezan sa  $x$

# Usmereni graf



Ivica  $(x, y)$  znači da je:

- $x$  povezan sa  $y$

# Neke definicije pojmova

**Stepen čvora (*degree*)** – broj čvorova koji su povezani sa datim čvorom.

*Usmereni graf definiše **ulazni stepen (*in-degree*)** i **izlazni stepen (*out-degree*)**.*

**Povezana komponenta (*connected component*)** – podskup čvorova koji su dostižni svaki iz svakog unutar datog podskupa i nemaju vezu sa ostalim čvorovima.

*Komponenta je **jako povezana** ako postoji putanja između svaka dva čvora u njoj.*

**Dvodelni graf (*bipartite graph*)** – graf kod kojeg se mogu definisati dva podskupa čvorova takvi da svaka ivica povezuje po jedan čvor iz svakog podskupa.

# Neke definicije pojmova

***Potpuno povezan neusmeren graf*** – graf koji ima ivicu između svakog para čvorova.

***Clique*** – potpuno povezan podgraf datog neusmerenog grafa.

***Hamiltonov ciklus*** – putanja koja povezuje sve čvorove grafa, prolazeći kroz svaki čvor tačno jednom.

# Predstava 1: Liste susedstva

*Adjacency List*: niz od  $V$  lista, takvih da:

$u \in V \rightarrow Adj[u]$  sadrži  $v$  takve da je  $(u, v) \in E$

Primer (ranije naveden neusmereni graf):

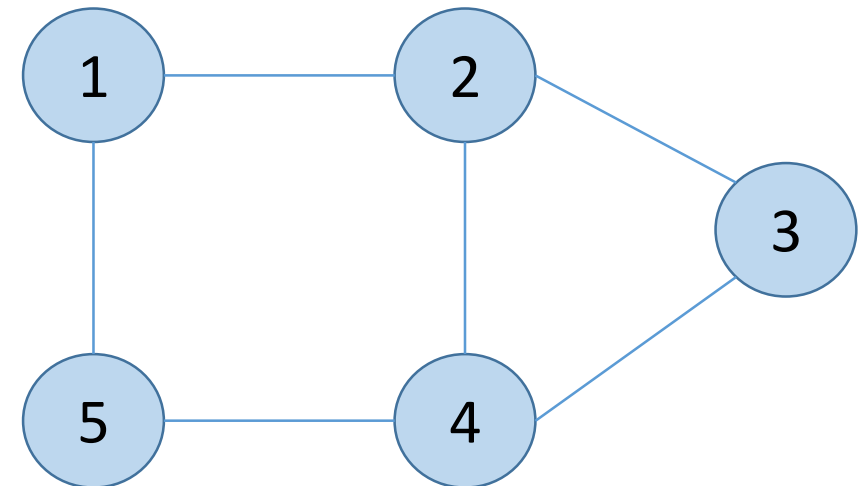
$1 \rightarrow 2, 5$

$2 \rightarrow 1, 3, 4$

$3 \rightarrow 2, 4$

$4 \rightarrow 2, 3, 5$

$5 \rightarrow 1, 4$





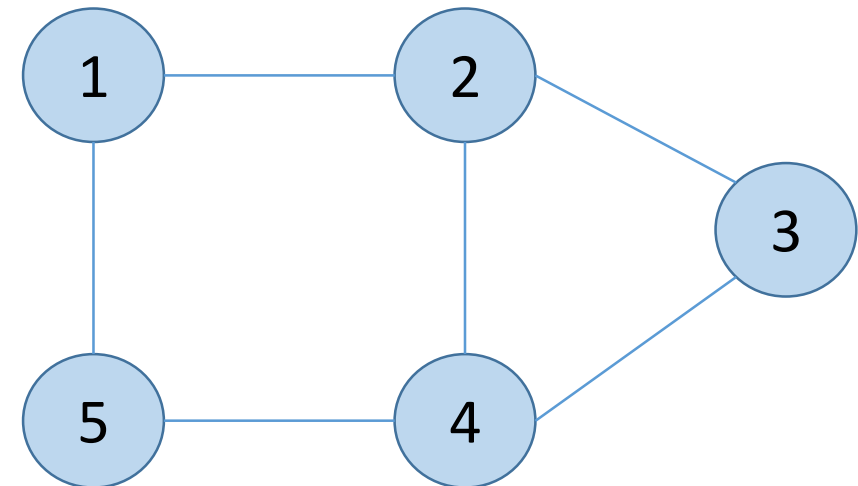
# Predstava 2: Matrica susedstva

*Adjacency Matrix:* matrica  $V \times V$  elemenata, takvih da:

$$a[i, j] = \begin{cases} 1, & (i, j) \in E \\ 0, & \textit{else} \end{cases}$$

Primer (ranije naveden neusmereni graf):

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



# Težinski grafovi

Težina  $w: E \rightarrow R$

broj koji definiše „težinu“ ivice, tj. vrednost koja se pridružuje svakoj ivici

Ako je graf prikazan kao lista, težina se može implementirati kao satelit podatak.

Ako je graf prikazan kao matrica, težina se može implementirati kao vrednost u matrici.

# Pretraga grafova – BFS

## **BFS – *Breadth-First Search*:**

- počinje od datog čvora  $s$
- otkriva sve čvorove koji su dostižni počev od čvora  $s$
- računa udaljenost svih dostižnih čvorova od čvora  $s$
- formira stablo (*Breadth-First Tree*) sa svim čvorovima dostižnim od čvora  $s$
- udaljenost do datog čvora je sigurno **najkraća udaljenost** od čvora  $s$

## Primena:

- nalaženje najkraće putanje
- analiza povezanosti grafa

# Pretraga grafova – BFS (priprema)

## **BFS (G, s)**

```
for each vertex  $u \in G.V - \{s\}$   
     $u.color = WHITE$   
     $u.d = Inf$   
     $u.p = NIL$   
 $s.color = GRAY$   
 $s.d = 0$   
 $s.p = NIL$   
 $Q = \emptyset$   
...
```

Color – boja čvora

- WHITE – čvor nije dostignut
- GRAY – čvor je dostignut
- BLACK – čvor i susedni čvorovi su dostignuti

d – udaljenost čvora od s

p – prethodni čvor u putanji od s

# Pretraga grafova – BFS (šetnja)

**BFS (G, s)**

...

ENQUEUE (Q, s)

**while**  $Q \neq \emptyset$

u = DEQUEUE (Q)

**for** each  $v \in G.Adj[u]$

**if**  $v.color == WHITE$

v.color = GRAY

v.d = u.d + 1

v.p = u

ENQUEUE (Q, v)

u.color = BLACK

Q – red čvorova koje smo dostigli,  
ali treba da obradimo (GRAY)

$$T(V, E) = O(V + E)$$

# Udaljenost čvorova

Udaljenost čvorova  $s$  i  $v$ :  $\delta(s, v)$  - dužina najkraće putanje od  $s$  do  $v$

Putanja – niz čvorova takvih da su:  $(s, a_1), (a_1, a_2), \dots, (a_{d-1}, v) \in E$

Neke osobine: ( $u$  je prethodni čvor od  $v$ )

$$\delta(s, v) \leq \delta(s, u) + 1$$

$$v.d = \delta(s, v)$$

# Pretraga grafova – DFS

## **DFS – *Depth-First Search*:**

- počinje od proizvoljnog čvora  $s$
- otkriva sve čvorove koji su dostižni počev od čvora  $s$
- iterativno nastavlja pretragu i na nedostignutim čvorovima, definišući novi  $s$
- formira šumu (*Depth-First Forest*) sa svim čvorovima grafa
- pamti dva vremenska trenutka: dostizanje čvora i završetak obrade čvora

## Primena:

- topološko sortiranje (sledeći čas)
- nalaženje putanje (npr. lavirint)
- provera povezanosti grafa

# Pretraga grafova – DFS

## DFS (G)

```
for each vertex  $u \in G.V$ 
     $u.color = WHITE$ 
     $u.p = NIL$ 
time = 0
for each vertex  $u \in G.V$ 
    if  $u.color == WHITE$ 
        DFS-VISIT( $G, u$ )
```

*color* – boja čvora

- WHITE – čvor nije dostignut
- GRAY – čvor je dostignut
- BLACK – čvor i susedni čvorovi su dostignuti

*p* – prethodni čvor u putanji od *s*

*time* – vreme (broj koraka)



# Pretraga grafova – DFS (jedna iteracija)

## **DFS-VISIT** ( $G, u$ )

```
time = time + 1
u.d = time
u.color = GRAY
for each  $v \in G.Adj[u]$ 
    if v.color == WHITE
        v.p = u
        DFS-VISIT( $G, v$ )
u.color = BLACK
time = time + 1
u.f = time
```

$d$  – vreme dostizanja čvora  
 $f$  – vreme završetka obrade čvora  
 $p$  – prethodnik čvora (roditelj)

$$T(V, E) = O(V + E)$$

# Osobine DFS

**Teorema zagrada** – ukoliko dostizanje čvora označimo otvorenom zagradom, a završetak obrade čvora zatvorenom zagradom, izraz je korektan sa stanovišta zagrada.

**Roditeljstvo** – ukoliko je  $v$  otkriven posle  $u$ , važi:  $u.d < v.d < v.f < u.f$

**Teorema bele putanje** – u šumi koja se formira pomoću DFS, čvor  $v$  je naslednik čvora  $u$  ako i samo ako u vremenu  $u.d$  u kome se otkrije čvor  $u$  postoji putanja od  $u$  do  $v$  koja sadrži samo bele čvorove.

# Osobine ivica grafa (pomoću DFS)

**Tree edge** – ivica  $(u, v)$  za koju važi da je  $v$  otkriven koristeći tu ivicu.

**Back edge** – ivica  $(u, v)$  za koju važi da je  $v$  prethodnik čvora  $u$ . Ovde spadaju  $(u, u)$ .

**Forward edge** – ivica  $(u, v)$  za koju važi da je  $v$  naslednik čvora  $u$ , ali ne direktan.

**Cross edge** – sve ostale ivice (između DFS stabala ili unutar DFS stabla).

Osobina neusmerenog grafa: **sve ivice su ili Tree ili Back ivice!**



thank you!

© Universal Studios, Revealing Homes