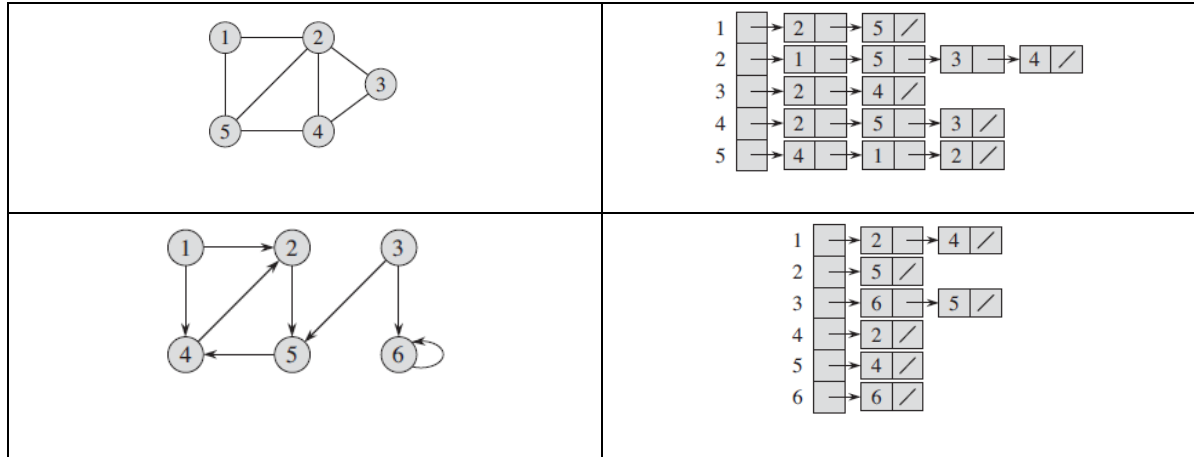


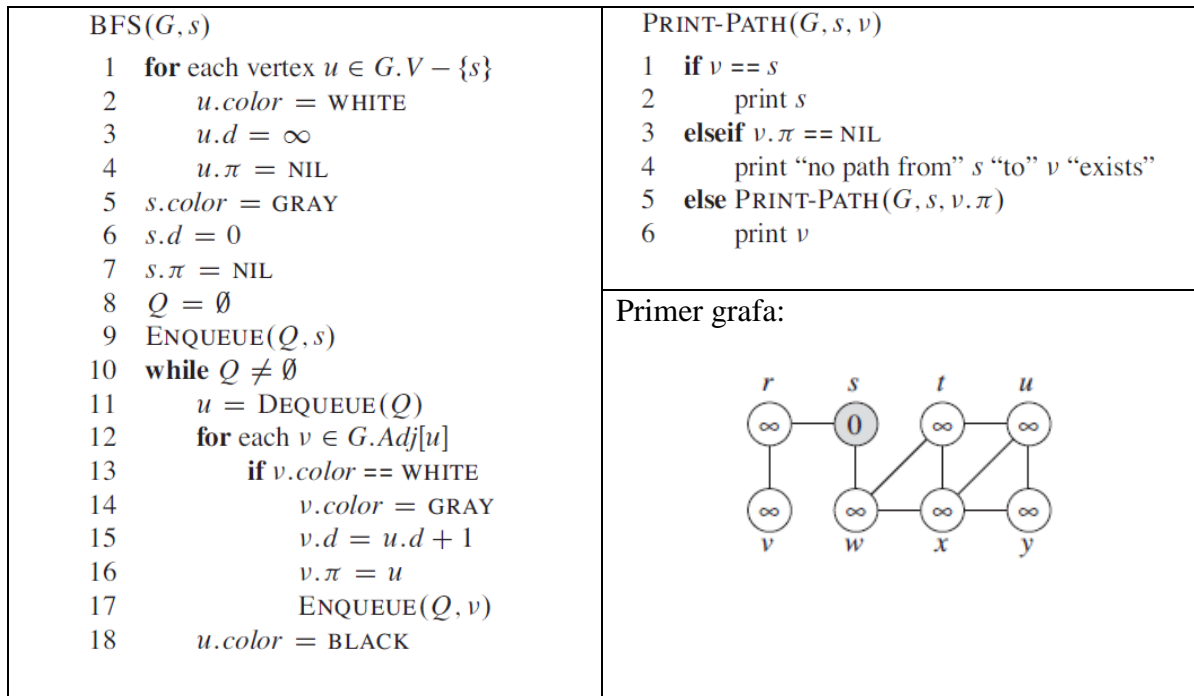
## Zadaci

- Implementirati grafove prikazane na slici 1. Graf predstaviti korišćenjem reprezentacije susednih lista čvorova (*adjacency-list representation*) koja je takođe prikazana na slici 1. Čvor grafa predstaviti po uzoru na priloženi šablon. Napisati funkcije za ispis svih susednih čvorova  $v$  čvora  $u$ .



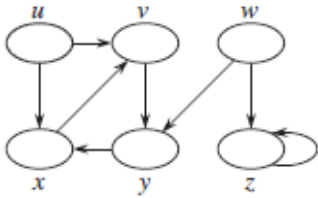
Slika 1 – Ilustracija usmerenog i neusmerenog grafa (levo) i implementacija istih upotrebom listi (desno).

- Implementirati pretragu grafa po širini (*Breadth-first search*). Za proveru funkcionalnosti iskoristiti graf sa slike 2. Takođe, na slici 2 dati su pseudokodovi funkcija.



Slika 2 – Pseudokod funkcija za pretragu grafa po širini.

3. Implementirati pretragu grafa po dubini (*Depth-first search*). Za proveru funkcionalnosti iskoristiti graf sa slike 3. Takođe, na slici 3 dati su pseudokodovi funkcija.

<pre> DFS(<i>G</i>) 1  for each vertex <i>u</i> ∈ <i>G.V</i> 2      <i>u.color</i> = WHITE 3      <i>u.π</i> = NIL 4  <i>time</i> = 0 5  for each vertex <i>u</i> ∈ <i>G.V</i> 6      if <i>u.color</i> == WHITE 7          DFS-VISIT(<i>G, u</i>)         </pre>	<pre> DFS-VISIT(<i>G, u</i>) 1  // white vertex <i>u</i> has just been discovered 1  <i>time</i> = <i>time</i> + 1 2  <i>u.d</i> = <i>time</i> 3  <i>u.color</i> = GRAY 4  // explore edge (<i>u, v</i>) 4  for each <i>v</i> ∈ <i>G.Adj</i>[<i>u</i>] 5      if <i>v.color</i> == WHITE 6          <i>v.π</i> = <i>u</i> 7          DFS-VISIT(<i>G, v</i>) 8  // blacken <i>u</i>; it is finished 8  <i>u.color</i> = BLACK 9  <i>time</i> = <i>time</i> + 1 10 <i>u.f</i> = <i>time</i>         </pre>
<p>Primer grafa:</p>  <pre> graph TD     u((u)) --&gt; v((v))     u((u)) --&gt; x((x))     v((v)) --&gt; y((y))     w((w)) --&gt; y((y))     w((w)) --&gt; z((z))     z((z)) --&gt; z((z))         </pre>	

Slika 3 – Pseudokod funkcija za pretragu grafa po dubini.