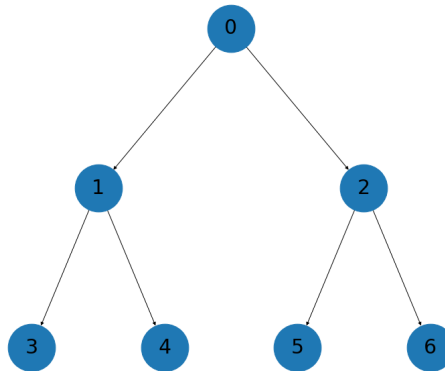


Questão 1

Considere o grafo a seguir



- (1) Dê uma representação por lista de adjacências do grafo.
- (2) Dê uma representação por matriz de adjacências equivalente do grafo.

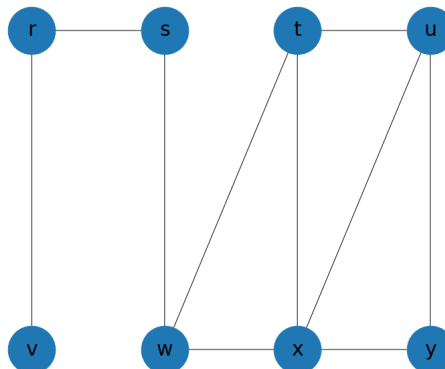
Questão 2

Dada uma representação por lista de adjacências de um grafo dirigido,

- (1) Qual o tempo necessário para calcular os graus de saída de todos os vértices?
- (2) Qual o tempo necessário para calcular os graus de entrada?

Questão 3

Considere o grafo a seguir

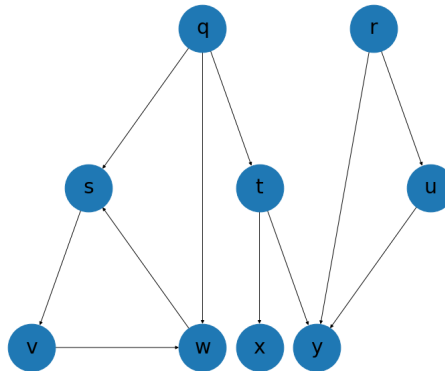


Encontre os valores d (atributo distância) e p (atributo predecessor) que resultam da execução da busca em largura(BFS) no grafo. Use o vértice u como fonte.

Obs: Veja Algorithm 1.

Questão 4

Considere o grafo a seguir



Executar a busca em profundidade no grafo. Suponha que os vértices são percorridos de acordo com a ordem alfabética.

Obs: Veja Algorithm 2 e Algorithm 3.

Questão 5

Dê um contra-exemplo para a seguinte hipótese: se um grafo dirigido G contém um caminho de u a v , então qualquer busca em profundidade deve resultar em $v.d \leq u.f$.

Questão 6

Reescreva o procedimento DFS(Algorithm 2) e DFS-VISIT(Algorithm 3) utilizando uma pilha para eliminar recursão.

Questão 7

Dê um contra-exemplo para a seguinte hipótese: se um grafo dirigido G contém um caminho de u a v e se $u.d < v.d$ em uma busca em profundidade de G , então v é um descendente de u na floresta em profundidade produzida.

Pseudo códigos dos algoritmos.

Algorithm 1 BFS(G, s)

```
1: for  $u \in V \setminus \{s\}$  do
2:    $u.cor \leftarrow BRANCO$ 
3:    $u.d \leftarrow \infty$ 
4:    $u.p \leftarrow NUL$ 
5: end for
6:  $s.cor \leftarrow VERMELHO$ 
7:  $s.d \leftarrow 0$ 
8:  $s.p \leftarrow NUL$ 
9:  $Q \leftarrow \emptyset$ 
10: ENQUEUE( $Q, s$ )
11: while  $Q \neq \emptyset$  do
12:    $u \leftarrow DEQUEUE(Q)$ 
13:   for  $v \in adj[u]$  do
14:     if  $v.cor == BRANCO$  then
15:        $v.cor \leftarrow VERMELHO$ 
16:        $v.d \leftarrow u.d + 1$ 
17:        $v.p \leftarrow u$ 
18:       ENQUEUE( $Q, v$ )
19:     end if
20:   end for
21:    $u.cor \leftarrow PRETO$ 
22: end while
```

Algorithm 2 DFS(G)

```
1: for  $u \in V$  do
2:    $u.cor \leftarrow BRANCO$ 
3:    $u.p \leftarrow NUL$ 
4: end for
5:  $time \leftarrow 0$ 
6: for  $u \in V$  do
7:   if  $u.cor == BRANCO$  then
8:     DFS-VISIT( $G, u$ )
9:   end if
10: end for
```

Algorithm 3 DFS-VISIT(G, u)

```
1:  $time \leftarrow time + 1$ 
2:  $u.d \leftarrow time$ 
3:  $u.cor \leftarrow VERMELHO$ 
4: for  $v \in adj[u]$  do
5:   if  $v.cor == BRANCO$  then
6:      $v.p \leftarrow u$ 
7:     DFS-VISIT( $G, v$ )
8:   end if
9: end for
10:  $v.cor \leftarrow PRETO$ 
11:  $time \leftarrow time + 1$ 
12:  $v.f \leftarrow time$ 
```
