

1) Descrivere come utilizzare la BFS per contare i vertici raggiungibili da s .

• Possibile soluzione 1:

ALGORITHMO BFS_STANDARD (G, s):

*** COUNT_REACHABLES = 0;**

// SETTING DEI NODI

FOR_EACH $v \in V \setminus \{s\}$:

| $v.color = white$;
| $v.\pi = NIL$;
| $v.d = \infty$;

Enqueue (Q, s);

// SETTING DELLA SORGENTE

$s.color = "gray"$;

$s.\pi = \emptyset$;

$s.d = 0$;

// ESPLORAZIONE

While ($Q.empty()$):

| $u = Q.dequeue()$;

| **FOR_EACH** v in $u.Adj[]$:

| | **if** $v.color == "white"$:

| | | **** COUNT_REACHABLES++;**

| | | $v.\pi = u$

| | | $v.color = "gray"$

| | | $v.d = u.d + 1$;

| | **Enqueue** (Q, v);

| $u.color = "Black"$;

COUNT_REACHABLE_VERTICES (G, s):

COUNT_REACHABLES = 0;

BFS_STANDARD (G, s);

FOR_EACH $v \in V \setminus \{s\}$:

| **if** ($v.d \neq \infty$):

| | **COUNT_REACHABLES++**

RETURN **COUNT_REACHABLES**;

• Possibile soluzione 2:

MODIFICO L'ALGORITHM CON * E **.

2) Scrivere un algoritmo che, dati un grafo $G=(V,E)$ **non orientato** ed un vertice $v \in V$, stampa l'elenco dei vertici che hanno distanza pari da v .

HO ANCORA 2 ALTERNATIVE:

- modificare direttamente l'algoritmo
- analizzare il suo output.

BFS (G, s):

FOR-EACH $v \in V$ / $d \neq \infty$:

$v.color = "white";$
 $v.\pi = NIL;$
 $v.d = \infty;$

Enqueue (Q, s);

$s.color = "gray";$
 $s.\pi = \emptyset;$
 $s.d = 0;$

*** PRINT (s);**

While ($isNotEmpty(Q)$) :

$u = Dequeue(Q);$

For_each ($v \in Adj[u] / u.\pi$) :

If ($v.color == "white"$) :

$v.color = "gray";$
 $v.\pi = u;$
 $v.d = u.d + 1;$

If ($v.d \% 2 == 0$) :

**** PRINT (v);**

Enqueue (Q, v);

$u.color = "black";$

}

Soluzione alternativa (BREVE):

PRINT-EVEN-DISTANCE (G, s) :

BFS (G, s);

FOR ($v \in V$) {

IF ($v.d \% 2 == 0$) :

PRINT (v)

}

}

}