

```

// visitare tutti i nodi a distanza  $k$  prima di visitare i nodi a distanza  $k+1$ 
bfs(GRAPH  $G$ , NODE  $r$ )
    QUEUE  $S \leftarrow$  Queue // creo una pila
     $S.enqueue(r)$  // inserisco la radice

    // inizializzazione
    bool[]  $visitato \leftarrow$  bool[1... $G.n$ ] // della dimensione del no. di nodi
    per ciascun  $u \in G.V - \{r\}$  fai  $visitato[u] \leftarrow$  falso // devo ancora visitarli
     $visitato[r] \leftarrow$  vero // radice visitata

    // visita del grafo
    finché not  $S.isEmpty$  fai
        NODE  $u \leftarrow S.dequeue$  // rimuovo un nodo

        { esamina il nodo  $u$  }

        per ciascun  $v \in G.adj(u)$  fai // per ciascun nodo adiacente "v"
            { esamina l'arco  $(u, v)$  }

            se not  $visitato[v]$  allora // se non ho ancora visitato "v"
                 $visitato[v] \leftarrow$  vero // marcalo come visitato
                 $S.enqueue$  // inseriscilo nella coda

```