

Si effettuano n visite in ampiezza, una a partire da ogni nodo, e si memorizza il massimo valore di distanza trovato in una visita.

```
// calcola il \più lungo cammino breve"
diametro(GRAPH G)
    int max ← 0
    QUEUE S ← new Queue
    S.enqueue(r)

    // inizializzazione
    bool[] dist ← new bool[1...G.size]
    per ciascun  $u \in G.V - \{r\}$  fai
        dist[u] ← -1

    dist[r] = 0 // la radice dista 0 da sé stessa
    finché not S.isEmpty fai
        NODE  $u \leftarrow S.dequeue$  // estrai il nodo
        per ciascun  $v \in G.adj(u)$  fai
            se  $dist[v] < 0$  allora
                // non ho ancora scoperto il nodo
                dist[v] ← dist[u] + 1 // imposta la distanza
            se  $dist[v] > max$  allora
                max = dist[v] // aggiorna il max
            S.enqueue(v) // inserisci il nodo
```