

```
// il cammino più breve fra due vertici viene memorizzato tramite il vettore dei padri p  
erdos(GRAPH G, NODE r, int[] erdos, NODE() parent)  
    QUEUE S ← Queue  
    S.enqueue(r) // inserisco la radice  
  
    // inizializzazione  
    bool[ ] visitato ← bool[1...G.n]  
    per ciascun  $u \in G.V - \{r\}$  fai  $erdos[u] \leftarrow \infty$   
  
     $erdos[r] \leftarrow$  vero // erdős ha distanza 0 da se stesso  
    parent[r] ← nil  
  
    finché not  $S.isEmpty$  fai  
        NODE u ← S.dequeue  
        per ciascun  $u \in G.adj(u)$  fai  
            { esamina l'arco  $(u, v)$  }  
            se  $erdos[v] \leftarrow \infty$  allora  
                // il nodo non è stato ancora scoperto  
                 $erdos[v] \leftarrow erdos[u] + 1$  // gli assegno un livello di erdős+1  
                parent[v] ← u // memorizzo il figlio nel vettore dei padri  
                S.enqueue(v) // è la prima volta che lo raggiungo quindi lo metto in coda
```