

```

// il cammino più breve fra due vertici viene memorizzato tramite il vettore dei padri p
erdos(GRAPH G, NODE r, int[] erdos, NODE() parent)

    // struttura di supporto
    QUEUE S ← Queue // creo una pila
    S.enqueue(r) // inserisco la radice

    // inizializzazione
    bool[] visitato ← bool[1...G.n]
    per ciascun  $u \in G.V - \{r\}$  fai erdos[u] ← ∞ // nodi non ancora raggiunti
    erdos[r] ← vero // erdős ha distanza 0 da se stesso
    parent[r] ← nil // per la stampa del cammino

    // visita del grafo
    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] ← ∞ allora
                // il nodo non è stato ancora scoperto

                erdos[v] ← erdos[u] + 1 // gli assegno un livello di erdős+1
                parent[v] ← u // memorizzo il padre del nodo attuale nel v. dei padri
                S.enqueue(v) // è la prima volta che lo raggiungo quindi lo metto in coda

```