Algoritmo 0.1 – Algoritmo generico

```
int, int CamminiMinimi(GRAPH G, NODE s)
    // Inizializzazione dei vettori
    int[ ] d ← new int[1...G.n]                    // distanze dalla sorgente
    int[ ] T ← new int[1...G.n]                    // vettore dei padri
    bool[ ] b ← new bool[1...G.n]   // per sapere in tempo costante se
     un nodo appartiene alla struttura dati

    // Inizializzo tutti i nodi tranne la sorgente
    per ciascun u ∈ G.V – {s} fai
        T[u] ← nil  // non hanno padri
        d[u] ← +∞  // non li ho ancora raggiunti
        b[u] ← falso  // non appartengono ancora all'insieme

    // Inizializzo la sorgente
    T[s] ← nil // non ha padre
    d[s] ← 0 // per convenzione
    b[s] ← vero // appartiene all'insieme
(1) STRUTTURADATI S ← StrutturaDati
    S.aggiungi (s)

    finché not S.isEmpty fai
(2)     int u ← S.estrai  // estraggo un nodo
        b[u] ← falso  // non è più contenuto nella struttura dati

        per ciascun v ∈ G.adj(u) fai // per tutti i vicini
            se d[u] + G.w(u,v) < d[v] allora // se migliora la stima
                se not b[v] allora // se non fa già parte
                 dell'insieme
(3)                 S.aggiungi(v) // aggiungilo
                    b[v] ← vero // fa parte dell'insieme
                altrimenti
(4)                 // Azione da intraprendere nel caso v sia già
                     presente in S

                // aggiorno i vettori
                T[v] ← u
                d[v] ← d[u] + G.w(u,v)

    ritorna (T, d)
```