

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 $u \in G.V - \{s\}$ 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 $S \leftarrow$ StrutturaDati
 $S.\text{aggiungi}(s)$

(2) finché not $S.\text{isEmpty}$ fai

```
int  $u \leftarrow S.\text{estrai}$  // estraigo un nodo
b[u] ← falso // non è più contenuto nella struttura dati
```

per ciascun $v \in G.\text{adj}(u)$ 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
         $S.\text{aggiungi}(v)$  // aggiungilo
        b[v] ← vero // fa parte dell'insieme
    altrimenti
        // 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)