```
(int[ ], int[ ]) CamminiMinimi(GRAPH G, NODE s)
    PRIORITYQUEUE S ← PriorityQueue  // O(n) · 1
    S.inserisci(s, 0)

    while not S.isEmpty do // O(n)
        // O(n) vettore ordinato / O(log n) heap binario
        int u ← S.deleteMin  b[u] ← false

        foreach v ∈ G.adj(u) do
            if d[u] + G.w(u,v) < d[v] then

                if not b[v] then
                    // O(1) · n vettore ordinato / O(log n) · n heap binario
                    S.inserisci(v, d[u] + G.w(u,v))   b[v] ← true
                else
                    // O(1) · m vettore ordinato / O(log n) · m heap binario
                    S.decrease(v, d[u] + G.w(u,v))

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

    return (T, d)
```