

```

greedyTsp(GRAPH G)
    SET S ← Set
    MFSET M ← Mfset(G.n)
    // archi in ingresso ad un nodo
    int[ ] in ← new int[1...G.n]

    // inizializzazione
    da i ← 1 fino a G.n fai
        in[i] ← 0

    { ordina gli archi per peso decrescente }

    per ciascun  $[u, v] \in G.E$  fai
        se  $in[u] \neq 2$  and  $in[v] \neq 2$  and  $M.\text{find}(u) \neq M.\text{find}(v)$  allora
            // non si è formato un ciclo
            S.insert( $\langle u, v \rangle$ )
            in[u]++
            in[v]++
            M.merge( $u, v$ )

    // comment
    int u ← 1
    finché  $in[1] \neq 1$  fai  $u++$ 
    int v ←  $u + 1$ 
    finché not  $in[v] \neq 1$  fai  $v++$ 

    // chiusura del circuito hamiltoniano
    S.insert( $\langle u, v \rangle$ )
    ritorna S

```