

Esercizio 1. Dato un albero T di n nodi, rappresentato tramite il vettore dei padri P , dare lo pseudo-codice di un algoritmo che produce in tempo $O(n)$ la lista dei vertici di T_u , il sotto-albero radicato in un vertice u di T .

SottoAlbero($P:array, u:nodo$) {

$n = P.length()$

$Sol: list$

$S: stack$

$Sot[n] = \{0, 0, \dots, 0\}$

$Sot[u] = 1$

$Sol.add(u)$

$k = \text{radice di } P \text{ // si trova in } O(n)$

 if ($k == u$) { return $\{0, 1, \dots, n\}$ }

 else { $Sot[k] = -1$ }

 for ($i = 0, 1, \dots, n$) {

$x = i$

 while ($Sot[x] == 0$) { // finche' trovo un elemento per cui so se e' o non e' in T_u

$S.push(x)$

$x = P[x]$

 }

$value = Sot[x] \text{ // } Sot[x] == 1 \Leftrightarrow \text{nello stack ci sono elementi di } T_u$

 while ($S \neq \emptyset$) {

$w = S.pop()$

$Sot[w] = value$

 if ($value == 1$) { $Sol.add(w)$ }

 }

 }

 return Sol

}

L'idea è quella di partire da un nodo e risalire verso la radice, appena trovo un nodo che fa parte del sotto-albero di u , tutti quelli dalla quale si è risaliti saranno anche essi nel sotto-albero. Se trovo un nodo che sicuramente non è nel sotto-albero, tutti quelli dalla quale si è risaliti non lo saranno. Alla fine tutti i nodi vengono visitati 2 volte. All'inizio, u è l'unico nel sottoalbero, e la radice l'unica che non vi è (a meno che la radice non sia proprio u , in tal caso la soluzione è costituita da tutti i nodi.)