

Esercizio 1. Dare lo pseudo codice di un algoritmo  $O(n)$  che accetta in input il vettore dei padri  $P[]$  di un albero e vertice  $x, y, z$  e restituisce il minimo antenato comune dei vertici  $x, y, z$ .

Prop: Se  $LCA(u, v) = x$ , allora  $LCA(u, v, w) = LCA(x, w)$

```

LCA(u:nodo, v:nodo, P:array) {
    AV[n] = {0, 0, ... 0}
    do {
        AV[u] = 1
        u = P[u]
        while (u != P[u])
        while (true) {
            if (AV[v] == 1) { return v }
            v = P[v]
        }
    }
}
//O(n)

```

```

LCA_glob(w:nodo, u:nodo, v:nodo, P:array) {
    x = LCA(u, v, P)
    return LCA(x, w, P)
}

```

Esercizio 2. Sia  $A[1, \dots, n]$  un array di  $n$  interi con  $n$  dispari. Sia  $A$  ordinato e ogni valore occorre esattamente due volte tranne uno. Dare il pseudocodice per un algoritmo  $O(\log n)$  che prenda in input l'array  $A$  e trova il valore presenta una sola volta. Per esempio, dato l'input:  $A = [2, 2, 5, 3, 7, 9, 10, 10]$ , l'algoritmo restituisce "7". Dato l'input:  $A = [1, 1, 4, 4, 6, 6, 8, 8, 9]$ , l'algoritmo restituisce "9".

```

Es2(A:array) {
    n = A.length()
    i =  $\lfloor \frac{n}{2} \rfloor$ 
    if (A[i] != A[i+1] & A[i] != A[i-1]) { return A[i] }
    if ((A[i] == A[i-1] & i % 2 == 0) || (i % 2 == 1 & A[i] != A[i-1])) {
        return Es2(A[0:i])
    }
    return Es2(A[i:n-1])
}

```