

```

TREE huffman(int[ ] c, int[ ] f, int n)
    // c[1...n]: caratteri dell'alfabeto
    // f[1...n]: frequenze dei caratteri
    // n: dimensione dell'alfabeto

    PRIORITYQUEUE Q ← MinPriorityQueue

    da i ← 1 fino a n fai //
         $\lfloor$  Q.inserisci( $f[i]$ , Tree( $f[i]$ ,  $c[i]$ )) //

    da i ← 1 fino a  $n - 1$  fai // n: radice  $\mathcal{O}(n)$ 
        // estraggo i 2 caratteri meno frequenti
         $z_1 \leftarrow Q.\text{deleteMin}$ 
         $z_2 \leftarrow Q.\text{deleteMin}$ 

        // Creo un nuovo nodo
         $z \leftarrow \text{Tree}(z_1.f + z_2.f, \text{nil})$ 
         $z.left \leftarrow z_1$ 
         $z.right \leftarrow z_2$ 

        // Lo inserisco nella coda
        Q.inserisci( $z.f$ ,  $z$ )  $\mathcal{O}(\log n)$ 

    ritorna Q.deleteMin

```