In the pseudo code that follows , we assume that C is a set of n characters and that each character c C is an object with an attribute c.freq giving its frequency. The algorithm builds the tree T corresponding to the optimal code in a bottom-up manner. It begins with a set of C leaves and performs a sequence of C 1 merging operations to create the final tree. The algorithm uses a min-priority queue Q, keyed on the freq attribute, to identify the two least-frequent objects to merge together. When we merge two objects, the result is a new object whose frequency is the sum of the frequencies of the two objects that were merged.

Algorithm Huffman Coding

```
$n := |C|$
$Q := C$
**for** $j \leftarrow 1$ **to** $n - 1$ **do**
    $allocate\ \ a\ \ new\ \ node\ \ z$
    $z.left := x := Extract - Min(Q)$
    $z.right :=:= Extract - Min(Q)$
    $z.freq := x.freq + y.freq$
    $Insert(Q, Z)$

**return** $Extract - Min(Q)$
```