CS3000: Algorithms & Data — Summer '24 — Laney Strange

Huffman Code (Greedy)

Below is the pseudocode for a Greedy implementation of the Huffman Code compression algorithm. It takes in a set C of n characters. We assume that every character $a \in C$ has a an attribute $a.freq$ indicating its total frequency in the starting file. It constructs and returns a tree, an optimal solution.

```
HUFFMAN(C)
 1  n = |C|
 2  Q = C
 3  for i = 1 to n − 1
 4       allocate a new node z
 5       x = EXTRACT-MIN(Q)
 6       y = EXTRACT-MIN(Q)
 7       z.left = x
 8       z.right = y
 9       z.freq = x.freq + y.freq
10       INSERT(Q, z)
11  return EXTRACT-MIN(Q)
```

Here's how we typeset the above function in CLRS style:

```
\begin{codebox}
\Procname{$\proc{Huffman}(C)$}
\li $n \gets |C|$
\li $Q \gets C$
\li \For $i \gets 1 \To n-1$
\Do
\li allocate a new node $z$
\li $x \gets \proc{Extract-Min}(Q)$
\li $y \gets \proc{Extract-Min}(Q)$
\li $z.\id{left} \gets x$
\li $z.\id{right} \gets y$
\li $z.\id{freq} \gets x.\id{freq} + y.\id{freq}$
\li $\proc{Insert}(Q,z)$
\End
\li \Return $\proc{Extract-Min}(Q)$
\end{codebox}
```