CPSC320 Assignment 3

Evan Louie
72210099
m6d7

1. The Huffman coding algorithm can create a prefix tree in O(n) time if the symbols are sorted by the probability in which they occur; Frequency of occurrence is in itself a measurement of probability.

   We will use 2 queues;

   a. The first one containing the initial frequencies (along with pointers to their associated leaves)
   b. The second will contain the combined frequencies after joining (along with pointers to their associated trees). This assures the lowest frequency is always kept at the front of one of the two queues.

   So in this example, we start by:

   - Enqueue all the leaf nodes into the first queue in increasing order (so f_1 will be the first followed by f_2 all the way to f_m, that is; in such a way that f_1 will be at the head of the queue)
   - While there exists more than 1 node in either of the queues:
     - Dequeue the 2 lowest frequency nodes. (By checking the front of each queue)
     - Create a new node with the just dequeued nodes as children and the sum of their frequencies as the new frequencies.
     - Enqueue this new node into the other queue
   - The remaining single node will be the root node.
   - The prefix tree has just been generated in O(n) time.

   Note: This example only works because the pairs were already sorted by frequency. As sorting would be a necessary action if unsorted, this would actually be slower than the standard Huffman algorithm as sorting in itself is O(nlogn).