COMP 3711H – Fall 2014
Tutorial 5

1. Huffman Coding
   From the book Problems on Algorithms, *by Ian Parberry, Prentice-Hall, 1995.*
   Build a Huffman Tree on the frequencies $\{1, 3, 5, 7, 9, 11, 13\}$.
   Are there any *ties* in the Huffman Construction process?
   How many *different* Huffman Trees can be built on this frequency set?

2. A Huffman Coding Variant

   (a) Recall that in each step of Huffman's algorithm, we merge two trees with the lowest frequencies. Show that the frequencies of the two trees merged in the ith step are at least as large as the frequencies of the trees merged in any previous step.

   (b) Suppose that you are given the n input characters, already sorted according to their frequencies. Show how you can now construct the Huffman code in $O(n)$ time. (*Hint:* You need to make clever use of the property given in part (a). Instead of using a priority queue, you will find it advantageous to use a simpler data structure.)

3. (CLRS–16.2-5) A *unit-length closed interval* on the real line is an interval $[x, 1 + x]$. Describe an $O(n)$ algorithm that, given input set $X = \{x_1, x_2, \ldots, x_n\}$, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct. You should assume that $x_1 < x_2 < \cdots < x_n$.

4. Huffman Coding and Mergesort
   Recall that Mergesort can be represented as a tree,
   with each internal node corresponding to a merge of two lists.
   The weight of a leaf is 1,
   with the weight of an internal node being the sum of the weights of its two children,
   or equivalently, the number of leaves in its subtrees.
   The cost of a single Merge is the number of items being merged,
   so the cost of Mergesort is the sum of the weights of the tree's internal nodes.

   (a) Prove that the cost of Mergesort can be rewritten as the weighted external path length of its associated tree, when all leaves have weight 1

   (b) Prove that the recursive Mergesort studied in class has height $h = \lceil \log_2 n \rceil$, with $x = 2^h - n$ leaves on level $h - 1$ and $n - x$ leaves on level h.

   (c) Show that an optimal Huffman Tree for n items all with the same frequency 1 can be built such that the tree will have height $h = \lceil \log_2 n \rceil$, with $x = 2^h - n$ leaves on level $h - 1$ and $n - x$ leaves on level h.

   (d) Use the above facts to prove that recursive mergesort is *optimal*, i.e., that there is no other merge pattern for merging n items that has lower total cost.