CS302 - Problem Set 10

1. Suppose you would like to use Huffman's Algorithm to encode 2^n different letters. Suppose you also know that the probability of the least frequent letter is more than half the probability of the most frequent letter. (Note: a perfect binary tree is a binary tree where all leaves have the same depth, so there are 2^k leaves for some $k \in \{0, 1, 2, \dots\}$).

   (a) **[6 points]** The following example has the property that the probability of the least frequent letter is more than half the probability of the most frequent letter, and there are 2^2 letters. Please create a Huffman Tree from this example, and describe the structure of the tree.

   $$A : .18, \quad B : ..2, \quad C : .26, \quad D : .36 \tag{1}$$

   (b) **[11 points]** Prove the following loop invariants hold for Huffman's algorithm, if the probability of the least frequent letter is more than half the probability of the most frequent letter, and there are 2^n letters.

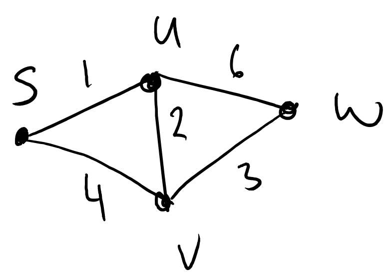
      i. If there are at least two trees, the probability of the least frequent tree is more than half the probability of the most frequent tree.

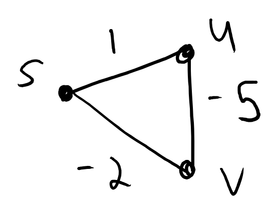
      ii. The new tree we create at a given iteration has a new effective probability that is larger than the probability of all current trees.

      iii. If there are at least two trees, and we order trees by probability, there is some $k \in \{0, 1, 2, \dots\}$ such that there are at least two perfect binary trees of depth k, possibly followed by some number of perfect binary trees of depth $k + 1$.

   (c) **[6 points]** What can you conclude from your proof? Should you use Huffman's algorithm in this situation? What is simpler encoding you could use in this situation.

2. **[6 points each]** Consider the following graphs and how Dijkstra's algorithm behaves on these instances. For each graph, for each loop of Dijkstra's algorithm, write the vertices in X at the beginning of the loop, the value of $A[v]$ for the vertex v most recently added to X, and calculate Dijkstra's criterion for each edge from X to $V - X$. Assume that s is the starting node in both graphs. For example, for the graph on the left, after intialization we have $X = \{s\}$, $A[s] = 0$, and the edges from X to $V - X$ are (s, u) with criterion value 1, and (s, v) with criterion value 4. Does the algorithm find the shortest path for each vertex in each graph?





3. **[6 points each]** For each of the following statements regarding Dijkstra's algorithm, either explain why it is true (proof not required), or provide a counter example.

   (a) Consider a graph G that is directed, has negative edge weights, but no negative cycles (a negative cycle is a cycle where the sum of edge-weights in the cycle have negative value.) Then there will always be a vertex where the incorrect distance is calculated.

   (b) Consider a graph G that is directed, and that has a negative cycle that is reachable from s. Then there will always be a vertex where the incorrect distance is calculated.