CSCI3160: Regular Exercise Set 5

Prepared by Yufei Tao

Problem 1. Let $G = (V, E)$ be a connected undirected graph where every edge carries a positive integer weight. Divide V into arbitrary disjoint subsets $V_1, V_2, ..., V_t$ for some $t \geq 2$, namely, $V_i \cap V_j = \emptyset$ for any $1 \leq i < j \leq t$ and $\bigcup_{i=1}^{t} V_i = V$. Define an edge $\{u, v\}$ in E as a *cross edge* if u and v are in different subsets. Prove: a cross edge with the smallest weight must belong to a minimum spanning tree (MST).

Problem 2* (Kruskal's Algorithm). Let $G = (V, E)$ be a connected undirected graph where every edge carries a positive integer weight. Prove that the following algorithm finds an MST of G correctly:

algorithm
1. $S = \emptyset$
2. **while** $|S| < |V| - 1$
3. find the lightest edge $e \in E$ that does not introduce any cycle with the edges in S
4. add e to S
5. return the tree formed by the edges in S

Problem 3. Consider Σ as an alphabet. Recall that a *code tree* on Σ is a binary tree T satisfying both conditions below:

- Every leaf node of T is labeled with a distinct letter in Σ; conversely, every letter in Σ is the label of a distinct leaf node in T.
- For every internal node of T, its left edge (if exists) is labeled with 0, and its right edge (if exists) with 1.

Define an *encoding* as a function f that maps each letter $\sigma \in \Sigma$ to a non-empty bit string, which is called the *codeword* of σ. T *produces* an encoding where the code word of a letter $\sigma \in \Sigma$ is obtained by concatenating the bit labels of the edges on the path from the root to the leaf σ. Prove:

- The encoding produces by a code tree T is a prefix code.
- Every prefix code f is produced by a code tree T.

Problem 4. Let T be an optimal code tree on an alphabet Σ (i.e., T has the smallest average height among all the code trees on Σ). Prove: every internal node of T must have two children.

Problem 5* (Textbook Exercise 16.3-7). Consider an alphabet Σ of $n \geq 3$ letters with their frequencies given. The prefix code we construct using Huffman's algorithm is *binary* because each letter $\sigma \in \Sigma$ is mapped to a string that consists of only 0's and 1's. Now, we want the code to be *ternary*, namely, each letter $\sigma \in \Sigma$ is mapped to a string that consists of three possible characters: 0, 1, or 2. As before, the code must be a prefix code. Assuming n to be an odd number, give an algorithm to find an encoding with the shortest average length.