CS 180: Data Structures, Fall 2011
Homework 5

Due by the start of class on Monday, Dec. 12

1. Draw the treap that results when the following pairs are inserted, where we form a BST over the letters and a min heap over the numbers:
$(R, 5), (S, 9), (E, 7), (H, 1), (W, 13), (D, 8), (J, 2), (K, 4), (P, 11)$.

   (a) Draw the frequency array and Huffman tree for the following string: "dogs to not spot hot pots or cats".

   (b) Using the Huffman tree from the previous part, write the binary encoding of the message "spot soda too".

   (c) How many bits did your encoding of the message in the previous problem take? How many bits would a standard ASCII encoding have taken (assuming 8 bits per character)?

2. Problems R-9.7, R-9.8, R-9.9, R-9.10 on page 417 of the textbook.

   (a) Draw the 11-entry hash table that results from using the hash function $h(i) = (3i + 5) \mod 11$ to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, assuming collisions are handled by *chaining* (or inserting into a list when there are collisions).

   (b) What is the result of the previous part, assuming collisions are handled by linear probing?

   (c) Show the result from part (a) assuming collisions are handled by quadratic probing (up to the point where the method fails).

   (d) Show the result from part (a) assuming collisions are handled by double hashing using a secondary hash function $h'(k) = 7 - (k mod 7)$.

3. Give an algorithm to detect in $O(m + n)$ time if a graph contains a cycle.

4. Consider the adjacency matrix and adjacency list representations of a graph. Which choice would be better in each of the following situations? Justify your answer.

   (a) The graph has 10,000 vertices and 20,000 edges, and it is important to use as little space as possible.

   (b) The graph has 10,000 vertices and 20,000,000 edges, and it is important to use as little space as possible.

   (c) You need to answer queries to find if two vertices are adjacent as fast as possible, no matter how much space you use.

5. We saw several algorithms which compute the connected component that some vertex s belongs to, such as DFS and BFS. Describe (in psuedocode) an $O(m + n)$ time algorithm to compute *all* the connected components of an undirected graph with n vertices and m edges. (Hint: you can use a simple modification to an algorithm from class.)

6. A phone company, RT&T, has designed a network of n switching stations connected by m high-speed communication links. Each phone connects directly to one of the switching stations in his or her area. The engineers of RT&T developed a prototype video chat system on the phone, but in order for it to have acceptable image quality, the number of links to transmit video cannot exceed 4. Design an efficient algorithm that computes, for each station, the set of stations it can link to for a video chat.