Problem Set 6 for CS 170

Problem 1 [Greedy Business]

Companies with values $W_1, W_2, \ldots, W_n$ are merged as follows. The two least valuable companies are merged, forming a list of $n-1$ companies. The *value of the merge* is the sum of the values of the two merged companies. This continues until one supercompany remains. Let V equal the sum of the values of the merges. (For example, if the initial values are $(3,3,2,2)$, the merges yield $(3,3,2,2) \rightarrow (4,3,3) \rightarrow (6,4) \rightarrow (10)$ and $V = 4+6+10$). Argue that V is the minimum value achievable by sequences of pair-wise merges terminating in one supercompany.

Problem 2 [Scheduling]

A small photocopying company faces the following scheduling problem. Each morning they get a set of jobs from customers. Customer i's job takes t_i time to complete. Also assume that each customer is given a weight that represents his or her importance to the company. Let w_i denote the weight of the ith customer. Assume that the company has a single machine on which jobs can be run. Given a schedule (i.e., an ordering of jobs) for that machine, let C_i denote the finishing time of job i. We want to find a schedule that minimizes $\sum_{i=1}^{n} w_i C_i$. Design an efficient algorithm to solve this problem.

Problem 3 [Cell Phones]

Consider a long country road with houses scattered very sparsely along it. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible.

Problem 4 [Huffman Lengths]

(a) Prove that in the Huffman coding scheme, if some character occurs with frequency greater than 2/5, then there is guaranteed to be a codeword of length 1. Also prove that if all characters occur with frequency less than 1/3, then there is guaranteed to be no codeword of length 1.

(b) Codex Unlimited manufactures hardware to decode Huffman-encoded game files in an XBox when you play them over a network. Its decoding hardware stores an entire codeword in a register when it is looking for a match. The uncompressed game files contain characters from an alphabet of size n. What is the minimum required register width (number of bits) that can support all Huffman codes for n characters, no matter what the frequencies $f_1, f_2, \ldots f_n$ are? What relationship between the frequencies $f_1, f_2, \ldots f_n$ causes the worst case (longest codeword) to occur?

Problem 5 [Lempel-Ziv Recovery]

(a) Alice sends Bob a Lempel-Ziv encoded file. She forgets to send Bob the dictionary. Bob knows Alice uses k bits to represent her dictionary indices (codewords), and that she uses a dictionary of size n. Can Bob decode the file? If so, how?

(b) Assume that we use Lempel-Ziv with a dictionary of unbounded size. If we encode a binary sequence of 28 bits, how large can the dictionary get in the worst case, where the size of the dictionary is the number of codewords in it? How small can the dictionary be in the best case? Give one of the worst-case and one of the best-case sequences.