1. Suppose that we have a set of $n$ activities to schedule among a large number of lecture halls. We wish to schedule all the activities using as few lecture halls as possible. Give a greedy algorithm to determine which activity should use which lecture hall. Provide the required time in the worst case, and justify the efficiency of your algorithm.

2. We are given $n$ jobs $j_1, j_2, \ldots, j_n$, with known running time $t_1, t_2, \ldots, t_n$, respectively. We have a single processor. We want to schedule these jobs so as to minimize the average completion time. Give a greedy algorithm and verify the optimality. (We assume nonpreemptive scheduling, i.e., once a job is started, it must run to completion.)

3. Give a Huffman code for the following string:

ACCGGTCGAGTGCGCGGAAGCCGGCCGAA

Describe your tree, the codeword, and the number of bits required to encode the string.

4. (a) What is a Huffman code for the following set of frequencies, based on the first 8 Fibonacci number?

   $a:1$, $b:1$, $c:2$, $d:3$, $e:5$, $f:8$, $g:13$, $h:21$

(b) Generalize your answer to find the optimal code when the frequencies are the first $n$ Fibonacci numbers?

5. We use Huffman’s algorithm to obtain an encoding of alphabet \{a, b, c\} with frequencies $f_a$, $f_b$ and $f_c$. In each of the following cases, either give an example of frequencies $\{f_a, f_b, f_c\}$ that would yield the specified code, or explain why the code cannot possibly be obtained (no matter what the frequencies are)

(a) Code: \{0, 10, 11\}  
(b) Code: \{0, 1, 00\}   
(c) Code: \{10, 01, 00\}

6. Suppose we want to make change for $n$ cents, using the least number of coins of denominations 1, 10, and 25 cents. Consider the following greedy strategy: suppose the amount left to change is $m$; take the largest coin that is no more than $m$; subtract this coin’s value from $m$, and repeat. Either prove that this algorithm always outputs an optimal solution, or give a counterexample to prove that this algorithm can output a non-optimal solution.