1. (3 points) Suppose that the dimensions of the matrices $A$, $B$, $C$, and $D$ are $8 \times 5$, $5 \times 11$, $11 \times 6$, and $6 \times 9$ respectively, and that we want to parenthesize the product $ABCD$ in a way that minimizes the number of scalar multiplications. Find the "m" and "s" tables computed by MATRIX-CHAIN-ORDER to solve this problem, and show the optimal parenthesization.

2. (3 points) Construct an optimal binary search tree for the keys $A$, $B$, $C$, $D$, $E$, $F$ with respective search probabilities $0.15$, $0.23$, $0.08$, $0.20$, $0.21$, and $0.13$ by computing entries for the tables "A" and "r".

3. (4 points) Given two strings, $X=x_1x_2x_3\ldots x_m$ and $Y=y_1y_2y_3\ldots y_n$, the shortest common supersequence (SCS) is a minimum length string $Z$ such that both $X$ and $Y$ are subsequences of $Z$. For example, if $X=<ABCBA>$ and $Y=<BCAABAB>$, then $Z=<ABCAABABA>$ is a SCS of both $X$ and $Y$. This problem is closely related the longest common subsequence problem we did in class. Give a dynamic programming algorithm which given $X$ and $Y$ computes the length of the SCS of $X$ and $Y$. For full credit, your algorithm should run in $O(mn)$ time. HINT: Construct a table $c[0, \ldots m, 0, \ldots n]$ where $c[i, j]$ is the length of the SCS of $x_1 \ldots x_i$ and $y_1 \ldots y_j$.

3a. Explain your solution in 3-4 sentences.
3b. Describe the recursive solution.
3c. Compute the runtime of your algorithm
3d. Verify the correctness of your algorithm to compute the SCS of $<ABCA>$ and $<ACBACBA>$ (show the table).

4. (3 points) Given a value $N$, if we want to make change for $N$ cents, and we have infinite supply of each of $S = \{S_1, S_2, \ldots, S_m\}$ valued coins, how many ways can we make the change? The order of coins doesn’t matter.

For example, for $N = 4$ and $S = \{1,2,3\}$, there are four solutions: $\{1,1,1,1\}, \{1,1,2\}, \{2,2\}, \{1,3\}$. So output should be 4.
For $N = 10$ and $S = \{2, 5, 3, 6\}$, there are five solutions: $\{2,2,2,2\}, \{2,2,3,3\}, \{2,2,6\}, \{2,3,5\}$ and $\{5,5\}$. So the output should be 5.

5. (4 points) Prove that Huffman codes are optimal in the sense that they represent a string of symbols using the fewest bits among all binary prefix codes.

6. (3 points) Let $F$ be a text file consisting of 128 characters, each in the set $A = \{a; b; c; d; e; f; g; h\}$, and let $T$ be a Huffman encoding tree for $F$. The height of $T$ depends on the frequencies the characters of $A$ in $F$.
1. Assign a frequency to each character of $A$ such that the height of $T$ is maximum.
2. Assign a frequency to each character of $A$ such that the height of $T$ is minimum.
In each case, determine the length (number of bits) of the encoded file. Note that in each case, the sum of the frequencies of the characters must be 128.

7. (5 points) There are 'n' programs that are to be stored on a computer tape of total length 'X'. Associated with each program $i$ is length $L_i$. Clearly all the programs can be stored on the tape if and only if the sum of the lengths of the programs is at most $X$. We shall assume that whenever a program is to be scanned to on the tape, the tape is initially positioned at the front. Hence, if programs are stored in the order $program_1, program_2, program_3, \ldots, program_n$ the time $t_i$ needed to scan to program $i$ is proportional to how far along the tape $program_i$ is, which depends on the lengths of the programs before $program_i$ on the tape. For example, if the programs are stored on the tape in the following order: $program_4, program_1, program_2, program_3$; then the time $t_4$ needed to scan to the fourth program in the list (program3) is proportional to
L_4 + L_1 + L_2. If all programs are scanned to equally often (each program has the same chance of being scanned to, then the expected or mean retrieval time (MRT) is
\[
\frac{1}{n} \sum_{j \leq s} t_j
\]

In the optimal storage on tape problem, we are required to find a permutation (ordering) for the 'n' programs so that when they are stored on the tape the MRT is minimized.

7a. Give an example of three programs (n=3) with different integer lengths (L_3, L_2, L_1). Show all the permutations (orderings) possible to store the programs on the tape. Calculate the mean retrieval time for each permutation and find the optimal ordering. This should give you an indication of a possible greedy choice strategy.

7b. Discuss and prove optimal substructure for the optimal storage on tape problem.
7c. Discuss and prove the greedy choice property for the optimal storage on tape problem.