**Problem 1.** (20 = 10 + 10 points). Let \( y_1, y_2, \ldots, y_n \) be a sequence of integers. Compute a subsequence using the following algorithm that starts with an empty stack. \textsc{Push} and \textsc{Pop} are the standard stack operations, and \textsc{Top} returns a copy of the item at the top of the stack, just like \textsc{Pop}, except it does not remove that item from the stack.

\[
\begin{align*}
\text{\textsc{Push}(\infty);} & \\
\text{for } i = 1 \text{ to } n \text{ do} & \\
\quad & \text{while } \textsc{Top} \leq y_i \text{ do } \textsc{Pop} \text{ endwhile;} \\
& \text{\textsc{Push}(y_i)} \\
\text{endfor}.
\end{align*}
\]

At termination, the computed subsequence is stored on the stack.

(a) Interpret each integer \( y_i \) as a point \((i, y_i)\) in the plane. What geometric meaning has the computed subsequence?

(b) Show that the amortized cost of a single operation (processing an integer \( y_i \)) is constant.

**Problem 2.** (20 = 10 + 10 points). Construct the optimal Huffman tree and the corresponding optimal binary code for the following alphabet of ten letters with frequencies shown in parentheses:

- A(5), B(6), C(9), D(2), E(11), F(5), G(6), H(14), I(4), J(2).

[Make sure that the weight of the root is the sum of frequencies, which is 64.]

(a) Show the Huffman tree.

(b) Show the binary code in a table.

**Problem 3.** (20 = 10 + 10 points). Let \( p_0 = 5, p_1 = 6, p_2 = 7, p_3 = 1, p_4 = 10, p_5 = 2 \), and for \( 1 \leq i < 5 \) let \( X_i \) be a matrix with \( p_{i-1} \) rows and \( p_i \) columns. Let \( X = X_1 \cdot X_2 \cdot X_3 \cdot X_4 \cdot X_5 \).

(a) Determine the minimum number of elementary multiplications needed to compute \( X \).

(b) Show the optimum parenthesization for computing \( X \).

**Problem 4.** (20 = 8 + 7 + 5 points). Consider the problem of making change for \( n \) cents using the fewest number of coins. Assume that each coin’s value is an integer.

(a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution.

(b) Suppose that the available coins are in the denominations that are powers of \( d \) (\( d^0, d^1, d^2 \) and so on) for some integer \( d > 1 \). Show that the greedy algorithm always yields an optimal solution.

(c) Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every \( n \).

**Problem 5.** (20 = 5 + 5 + 5 + 5 points). Solve each of the following recurrences using the Master Method:

(a) \( T(n) = 4T(n/4) + n^2 \);

(b) \( T(n) = 3T(n/3) + n \);

(c) \( T(n) = 7T(n/4) + n^{1.5} \);

(d) \( T(n) = 9T(n/3) + n^2 \).