Problem 1

(5 points) A certain string-processing language offers a primitive operation which splits a string into two pieces. Since this operation involves copying the original string, it takes \( n \) units of time for a string of length \( n \), regardless of the location of the cut. Suppose, now, that you want to break a string into many pieces. The order in which the breaks are made can affect the total running time. For example, if you want to cut a 20-character string at positions 3 and 10, then making the first cut at position 3 incurs a total cost of 20+17=37, while doing position 10 first has a better cost of 20+10=30.

Give a dynamic programming algorithm that, given the locations of \( m \) cuts in a string of length \( n \), finds the minimum cost of breaking the string into \( m + 1 \) pieces.

Lay down the solution by explicitly following the steps we saw in class (Thursday April 8).

- What are the subproblems and their parameters? Is there a particular assignment of the parameters that make the subproblem equivalent to the original problem?
- Now check for the optimal substructure. If a certain subproblem is optimal is there a set of sub-sub-problems that are also optimal? Write down the recurrence.
- Build the dynamic programming table. What is the complexity?

Problem 2

(5 points) Consider a 2-player game using coins. At the start of the game, there is an even number of coins laid out in a row. Each coin has some value, \( v(1) \ldots v(n) \). We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Such games often benefit the player who goes first. Determine the maximum possible amount of money we can definitely win if we move first.

OPTIONAL EXTRA-CREDIT PROBLEM

(5 points) You have a set of \( n \) integers each in the range 0...\( K \). Partition these integers into two subsets, \( \text{(sums)} \) such that you minimize the absolute value of the difference between them. That is, \( |S1 - S2| \) should be as small as possible, where \( S1 \) and \( S2 \) denote the sums of the elements in each of the two subsets.