Dynamic Programming II

1. **Counting Boolean Parenthesizations**. You are given a boolean expression consisting of a string of the symbols ’true’, ’false’, ’and’, ’or’, and ’xor’. Count the number of ways to parenthesize the expression such that it will evaluate to true. For example, there is only 1 way to parenthesize ’true and false xor true’ such that it evaluates to true.
2. **Optimal Strategy for a Game**. Consider a row of n coins of values v1 . . . vn, where n is even. 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. Determine the maximum possible amount of money we can definitely win if we move first.
3. **Two-Person Traversal of a Sequence of Cities**. You are given an ordered sequence of n cities, and the distances between every pair of cities. You must partition the cities into two subsequences (not necessarily contiguous) such that person A visits all cities in the first subsequence (in order), person B visits all cities in the second subsequence (in order), and such that the sum of the total distances travelled by A and B is minimized. Assume that person A and person B start initially at the first city in their respective subsequences.
4. **Bin Packing (Simplified Version)**. You have n1 items of size s1, n2 items of size s2, and n3 items of size s3. You’d like to pack all of these items into bins each of capacity C, such that the total number of bins used is minimized.