1. You are climbing a stair case. It takes \( n \) steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top? Give a dynamic programming algorithm to solve this problem. Note: \( n \) is a positive integer.

2. You are given an array \( A = (a_1, a_2, ..., a_n) \) of \( n \) positive integers. The objective is to find a pair \( (a_i, a_j) \) such that \( i < j \) and \( d_j - d_i \) is maximum and positive. For example, for \( A = (9, 2, 4, 3, 8) \), a solution is \( (2, 8) \). For \( A = (9, 6, 5, 3, 1) \), there is no such a pair. Find a dynamic programming algorithm to solve this problem.

3. Given two strings \( a \) and \( b \), check if \( a \) is a subsequence (not necessarily consequent subsequence) of \( b \).
   Example:
   \[ a = \text{aegis76} \]
   \[ b = \text{evatookplanetohamburgonsomedayin1976} \]
   Then \( a \) is a subsequence of \( b \).
   Give a dynamic programming algorithm to solve this problem.

4. Given a string \( s \), find the longest palindromic subsequence (LPS) length in \( s \). Example:
   \[ s = \text{axbddd} \] the LPS length is 4. Give a dynamic programming algorithm to solve this problem.