Problems (must be written up and turned in; refer to or use a solution template to understand the required elements of your answer)

1. There is a robot on the line at coordinate 1. In one move, the robot can get from coordinate \( i \) to any one of the coordinates \( i + 1, i + 2, \ldots, i + k \). Give an algorithm to determine how many distinct ways the robot can get from coordinate 1 to coordinate \( n \), where \( n \) and \( k \) are positive integer numbers and \( k \leq n \). Your algorithm must have \( O(n) \) time complexity and \( O(k) \) space complexity.

2. You are given a set of integer numbers \( A = \{a_1, a_2, \ldots, a_n\} \), where \( 1 \leq a_i \leq m \) for all \( 1 \leq i \leq n \) and for a given positive integer \( m \). Give an algorithm which determines whether you can represent a given positive integer \( k \leq nm \) as a sum of some numbers from \( A \), if each number from \( A \) can be used at most once. Your algorithm must have \( O(nk) \) time complexity.

3. You are given a sequence of \( n \) numbers \( A = (a_1, a_2, \ldots, a_n) \). At one step, you can erase any number except for the leftmost and the rightmost ones. Erasing number \( a_i \) costs \( a_{i-1} \cdot a_{i+1} \). Your goal is to erase all the numbers \( a_i \) where \( 1 < i < n \), in some order, such that the total cost is minimized. Give an algorithm to accomplish your goal, with \( O(n^3) \) time complexity.

4. We consider some sequence \((b_1, b_2, \ldots, b_m)\) to be interesting if \( |b_{i+1} - b_i| < K \) for all \( 1 \leq i \leq m - 1 \), where \( K \) is a given positive number. You are given a sequence of \( n \) numbers: \((a_1, a_2, \ldots, a_n)\) and a number \( K \). Give an algorithm which finds the length of the longest interesting subsequence of the given sequence. The time complexity of your algorithm should be \( O(n^2) \).

5. You are given a two-dimensional array \( A \) of dimensions \( k \times n \). The elements of the array are integer numbers. Your goal is to find the sum of the following sequence \( D = (d_1, d_2, \ldots, d_n) \):

   \[(a) \quad d_i \in \{A[1][i], A[2][i], \ldots, A[k][i]\} \text{ for all } 1 \leq i \leq n, \text{ i.e., } d_i \text{ is some element in the array } A \text{ at column } i.

   \(b) \text{ Neighboring elements in } D \text{ can’t be from the same row in array } A. \text{ For example, if } d_i = A[x][i] \text{ for some } x, \text{ then } d_{i-1} \neq A[x][i-1] \text{ and } d_{i+1} \neq A[x][i+1].

   \(c) \text{ The sum of all elements in } D, \text{ i.e., } d_1 + d_2 + \ldots + d_n, \text{ is maximized. Your goal is to find the sum of the sequence } D.\]

Provide a solution with \( O(nk) \) time and \( O(k) \) space complexity for full credit.