1. (2 marks) List out data structures you learnt in this course, then explain each in detail.

|  |
| --- |
|  |

1. (2 marks) Describe listed algorithm design strategies: Brute Force, Decrease and Conquer, Divide and Conquer, Transform and Conquer, Greedy Technique, Graph Algorithm, dynamic programming.

|  |
| --- |
|  |

1. (3 marks) make your java implementation of Fibonacci function using dynamic programming technique, submit your java file.

Your main function should invoke fibo(3), fibo(7), fibo(5), and paste your output below

|  |
| --- |
|  |

1. (3 marks) **Problem Statement:**

Given an integer array nums, return the length of the longest strictly increasing subsequence.

**Java solution:**

|  |
| --- |
| public class LongestIncreasingSubsequence {  public static int lengthOfLIS(int[] nums) {  if (nums.length == 0) return 0;  int[] dp = new int[nums.length];  for (int i = 0; i < dp.length; i++) {  dp[i] = 1; // Each element is an increasing subsequence of at least length 1.  }  for (int i = 1; i < nums.length; i++) {  for (int j = 0; j < i; j++) {  if (nums[i] > nums[j]) {  dp[i] = Math.max(dp[i], dp[j] + 1);  }  }  }  // Find the maximum value in dp array  int maxLength = 0;  for (int length : dp) {  maxLength = Math.max(maxLength, length);  }  return maxLength; // The length of the longest increasing subsequence.  }  public static void main(String[] args) {  int[] nums = {10, 9, 2, 5, 3, 7, 101, 18};  System.out.println(lengthOfLIS(nums)); // Output: 4  }  } |

Your task is to explain why this solution is dynamic programming.

|  |
| --- |
|  |