

[Dashboard](#) / [My courses](#) / [CS23331-DAA-2023-CSE](#) / [Dynamic Programming](#) / [4-DP-Longest non-decreasing Subsequence](#)

Started on	Monday, 18 November 2024, 8:37 PM
State	Finished
Completed on	Monday, 18 November 2024, 8:37 PM
Time taken	21 secs
Marks	1.00/1.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Problem statement:

Find the length of the Longest Non-decreasing Subsequence in a given Sequence.

Eg:

Input:9

Sequence: [-1,3,4,5,2,2,2,3]

the subsequence is [-1,2,2,2,2,3]

Output:6

Answer: (penalty regime: 0 %)

```

1  #include <stdio.h>
2  int longest_non_decreasing_subsequence(int sequence[], int n) {
3      int dp[n];
4      for (int i = 0; i < n; i++) {
5          dp[i] = 1;
6      }
7      for (int i = 1; i < n; i++) {
8          for (int j = 0; j < i; j++) {
9              if (sequence[j] <= sequence[i]) {
10                 dp[i] = (dp[i] > dp[j] + 1) ? dp[i] : dp[j] + 1;
11             }
12         }
13     }
14     int max_len = dp[0];
15     for (int i = 1; i < n; i++) {
16         if (dp[i] > max_len) {
17             max_len = dp[i];
18         }
19     }
20     return max_len;
21 }
22 int main() {
23     int sequence[] = {-1, 3, 4, 5, 2, 2, 2, 3};
24     int n = sizeof(sequence) / sizeof(sequence[0]);
25     int result = longest_non_decreasing_subsequence(sequence, n);
26     printf("%d\n", result);
27     return 0;
28 }
29

```

	Input	Expected	Got	
✓	9 -1 3 4 5 2 2 2 2 3	6	6	✓
✓	7 1 2 2 4 5 7 6	6	6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◀ 3-DP-Longest Common Subsequence

Jump to...

1-Finding Duplicates- $O(n^2)$ Time Complexity, $O(1)$ Space Complexity ▶