Dashboard / My courses / CS23331-DAA-2023-CSE / Dynamic Programming / 4-DP-Longest non-decreasing Subsequence

Started on	Wednesday, 20 November 2024, 6:24 PM
State	Finished
Completed on	Wednesday, 20 November 2024, 6:27 PM
Time taken	3 mins 30 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:[-1,3,4,5,2,2,2,2,3]

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

Output:6

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

// Function to find the length of the longest non-decreasing subsequence
int lnds(int arr[], int n) {
    int dp[n];
    for (int i = 0; i < n; i++) {
        dp[i] = 1;
    }

    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (arr[i] >= arr[j] && dp[i] < dp[j] + 1) {
                dp[i] = dp[j] + 1;
            }
        }
    }

    int max = 0;
    for (int i = 0; i < n; i++) {
        if (dp[i] > max) {
            max = dp[i];
        }
    }

    return max;
}

int main() {
    int sequence[] = {-1, 3, 4, 5, 2, 2, 2, 2, 3};
    int n = sizeof(sequence)/sizeof(sequence[0]);
    printf("%d\n", lnds(sequence, n));
    return 0;
}

```

	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 ►