4-DP-Longest non-decreasing Subsequence

Aim:

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 %)

Algorithm:

1. Read the integer n and array sequence of size n.
2. Initialize a dp array where each element is initially 1, representing the length of the longest non-decreasing subsequence ending at each position.
3. Use nested loops to update dp[i] by checking if sequence[j] <= sequence[i] and updating dp[i] to the maximum of dp[i] and dp[j] + 1.
4. Track the maximum length found during the iterations and return it.
5. Print the result.

Code:

#include <stdio.h>

int longestNonDecreasingSubsequence(int n, int sequence[]) {

int dp[n];

int maxLength = 1;

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 (sequence[j] <= sequence[i]) {

dp[i] = (dp[i] > dp[j] + 1) ? dp[i] : dp[j] + 1;

}

}

maxLength = (maxLength > dp[i]) ? maxLength : dp[i];

}

return maxLength;

}

int main() {

int n;

scanf("%d", &n);

int sequence[n];

for (int i = 0; i < n; i++) {

scanf("%d", &sequence[i]);

}

int result = longestNonDecreasingSubsequence(n, sequence);

printf("%d", result);

}

Output:

|  | **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.

Result:

The expected output was obtained