Click to register for Certification exam (https://examform.nptel.ac.in/2025_01/exam_form/dashboard)

If already registered, click to check your payment status

Week 11 : Assignment 11

Assignment not submitted

Due date: 2025-04-09, 23:59 IST.

Common Statement for Question 1 & 2

Your final End term exams are going to be over and you are catching up on Netflix. You have a schedule of interesting live shows during the next day. You hate to start or stop watching a show midway, so your aim is to watch as many complete shows as possible during the day.

Suppose there are n such shows $M_1, M_2, \ldots, M_n$ available during the coming day. The shows are ordered by starting time, so for each $i \in 1, 2, \ldots, n-1$, M_i starts before M_{i+1}. However, show M_i may not end before M_{i+1} starts, so for each $i \in 1, 2, \ldots, n-1$, $Next[i$ is the smallest $j > i$ such that M_j starts after M_i finishes if exist, otherwise `-1` .

Given the sequence $M_1, M_2, \ldots, M_n$ and the values $Next[i]$ for each $i \in 1, 2, \ldots, n-1$, your aim is to compute the maximum number of complete shows that can be watched.

1) Let $Watch[i]$ denote the maximum number of complete shows that can be watched among $M_i, M_{i+1}, \ldots, M_n$. Which of the following is a correct formulation of $Watch[i]$ for $i \in n-1, n-2, \ldots, 2, 1$? Consider initially $Watch[n] = 1$. ***1 point***

- ◯ $$Watch[i] = \begin{cases} Watch[i+1], & if \;\; Next[i] = -1 \\ max(Watch[Next[i]], Watch[i+1]), & if \;\; Next[i] \neq -1 \end{cases}$$
- ◯ $$Watch[i] = \begin{cases} Watch[i+1], & if \;\; Next[i] = -1 \\ max(Watch[Next[i]], 1 + Watch[i+1]), & if \;\; Next[i] \neq -1 \end{cases}$$
- ◯

$$Watch[i] = \begin{cases} Watch[i+1], & if\ \ Next[i] = -1 \\ max(1 + Watch[Next[i]], Watch[i+1]), & if\ \ Next[i] \neq -1 \end{cases}$$

○
$$Watch[i] = \begin{cases} 1 + Watch[i+1], & if\ \ Next[i] = -1 \\ max(Watch[Next[i]], Watch[i+1]), & if\ \ Next[i] \neq -1 \end{cases}$$

2) How much time given dynamic programming approach will take to compute the answer? Assume you have direct access to the $Next$ list as well and you don't have to worry about computing it on your own. ***1 point***

○ $O(n^2)$

○ $O(n^3)$

○ $O(n\ log\ n)$

○ $O(n)$

Common Statement for Question 3 and 4

Longest strictly decreasing subsequence (continuous) refers to a specific subset of elements within a larger sequence that meet the following criteria:

1. **Strictly Decreasing:** Each element in the subsequence must be strictly **larger** than the element that follows it. There can't be any equal values.
2. **Continuous:** The elements in the subsequence must appear consecutively in the original sequence without any gaps. They must be neighbors in the original order.
3. **Longest:** The subsequence should have the maximum possible length among all such decreasing subsequences within the original sequence.

3) Consider the input array $A = [4, 3, 2, 6, 8, 7, 7, 5, 4, 2, 1]$
What is the length of the longest strictly decreasing subsequence(continuous) of numbers in array A?

1 point

4) Consider the following algorithm to find the length of the longest trictly decreasing subsequence(continuous) of numbers in array $A_{0\ldots n-1}$: ***1 point***

1. $n = length(A)$
2. $Initialize\ list\ L_{0\ldots n-1} = 0$
3. $L_0 = 1$
4. $For\ all\ i,\ start\ from\ index\ 1\ to\ n-1:$
5. **Inductive structure**
6. $return\ max(L)$

Note:- L_j is the length of the longest strictly decreasing sequences ending at A_j , where $0 \leq j \leq n-1$.
Which of the following is the correct **inductive structure** to fill at step 5 to return the correct result?

Quiz: Week 11 : Assignment 11 (assessment?name=554)

Week 12 ()

Download Videos ()

Live Sessions ()

Transcripts ()

Books ()

$$L_i = \begin{cases} 1 + L_{i+1}, & if \;\; A_i > A_{i+1} \\ 1, & Otherwise \end{cases}$$

$$L_i = \begin{cases} 1 + L_{i-1}, & if \;\; A_i > A_{i-1} \\ 1, & Otherwise \end{cases}$$

$$L_i = \begin{cases} 1 + L_{i-1}, & if \;\; A_i < A_{i-1} \\ 1, & Otherwise \end{cases}$$

$$L_i = \begin{cases} 1 + L_{i-1}, & if \;\; A_i \geq A_{i-1} \\ 1, & Otherwise \end{cases}$$

Common Statement for Question 5 and 6

Consider the following function `do_something(A,B)` , where A and B are two strings of length m and n respectively.

```
def do_something(A, B):
    m = len(A)
    n = len(B)
    T = [[0] * (n + 2) for i in range(m + 2)]
    for i in range(m + 1):
        for j in range( n + 1):
            if i == 0:
                T[i][j] = j
            elif j == 0:
                T[i][j] = i
            elif (A[i - 1] == B[j - 1]):
                T[i][j] = 1 + T[i - 1][j - 1]
            else:
                T[i][j] = 1 + min(T[i - 1][j], T[i][j - 1])
    return T[m][n]
```

5) What does function `do_something(A, B)` return? ***1 point***

- Length of the shortest possible string `s` , where `A` and `B` are subsequence of `s` .
- Length of the longest common subsequence of `A` and `B`
- Length of the longest continuous common sequence of `A` and `B`
- Edit distance between `A` and `B`

6) What is the time complexity of function `do_something(L, k)` ? ***1 point***

- $O(m+n)$
- $O(m\ log\ n)$
- $O(mn)$
- $O(n\ log\ m)$

7) Let you have a single positive integer x , which is initially equal to1. You are given a positive integer n where n > x .

In each step, you can either increment x by 1 or double x . Your goal is to produce a target value n in minimum number of steps. For example, you can produce the integer n = 10 in four steps as follows:

$$1(+1) \rightarrow 2(*2) \rightarrow 4(+1) \rightarrow 5(*2) \rightarrow 10$$

What is the minimum number of steps required if n = 1025 ?

1 point

You may submit any number of times before the due date. The final submission will be considered for grading.

Submit Answers