Luis Enrique Franco Marín

410921353

**Week 5: Longest Increasing Subsequence**

A longest increasing subsequence is a subarray of an array which has the largest possible number of numbers that are in increasing order. The same can be done for a subsequence which is in decreasing order. A Wavio sequence has the following characteristics:

* Wavio is of odd length i.e., L−2×n+1.
* The first (n+1) integers of Wavio sequence makes a strictly increasing sequence.
* The last (n+1) integers of Wavio sequence makes a strictly decreasing sequence.
* NO two adjacent integers are same in a Wavio sequence.

By these criteria a Wavio sequence is comprised of two subsequences of increasing and decreasing order respectively. Whit the use of dynamic programming storing the potential increasing sequences and replacing them with more optimal ones then a sequence can be constructed.

**Problem:**

The input file contains less than 75 test cases. The description of each test case is given below. Input is terminated by end of file.

Each file starts with a positive integer, N, (1≤N≤10000). In next few lines there will be N integers.

For each set of input print the length of longest Wavio sequence in a line.

**Code:**

Text

Description automatically generated

**Discussion:**

The problem is solved using dynamic programing, for this there are 4 extra sequences constructed, in the form of vectors. These sequences are used to store the increasing and decreasing subsequences of the sequence given.

By storing these sequences and then replacing them with sequences that are of longer length then the problem is solving iterating only one time through the array of N numbers.

Storing the array 4 extra times (potentially) gives the algorithm a space complexity of 4N which is **O(N).**

The array is iterated just once and the increasing and decreasing construction of the subsequences is work done inside this iteration. After this the vectors constructed are iterated potentially n times and the lower\_bound function has a complexity of logn so the time complexity of the algorithm is **O(NlogN)**.

**Result:**

A picture containing company name

Description automatically generated