**=> Longest Increasing subsequence in array of integers :**

O(n^2) : <https://ideone.com/JVufov>

Explanation : <https://www.youtube.com/watch?v=KoenMWrZmZU>

O(nlogn) time and O(1) space : <https://ideone.com/lKqL7B>

Explanation : <https://www.youtube.com/watch?v=TocJOW6vx_I>

**//Accepted on CSES :**

O(nlogn) time and O(n) space : <https://ideone.com/AUmrGd>

**Edge Cases : DUPLICATES**

5,1,8,2,12,3,4

Initialy LIS length = 1.

-> Now next element is 1, which is less than 5 , so that’s not increasing .

So replace first element in array (i.e=5 with 1),as any element further in array which is greater than 5 will also be greater than 1.

Array to be searched = 1

-> Next=8, it’s greater than 1,so replace 2nd element of array with 8.

Array to be searched = 1 8

-> Next = 2 , binary search for finding position of 2, it is after 1,so replace 8 with 1,as any element further in array which is greater than 8 will also be greater than 2.

Array to be searched = 1 2

-> Next =12 , it’s greater than 2,so replace 3rd element of array with 12

Array to be searched = 1 2 12

-> Next = 3, it’s real position is after 2 , so replace 12 with 3

Array to be searched = 1 2 3

-> Next =4 , it’s greater than 3, so replace 4th element of array with 4

Array to be searched = 1 2 3 4

Done, LIS length =4

**=> Count all LIS :**

**-> O(nlogn) version(Hard)**

**Problem :**[**https://leetcode.com/problems/number-of-longest-increasing-subsequence/**](https://leetcode.com/problems/number-of-longest-increasing-subsequence/)

**Approach(imp) :**

-> We have vector<vector<pair<int,int>>> **dp ,**

dp[1] will contain all numbers which form a subsequence of length=1,,same way dp[2],dp[3] and so on…

->The 2nd parameter stores the count of number of subsequences that will end at that number.

-> Now any new number which comes, we will compare it with the **smallest number** in all these vectors, for e.g if it is greater than the smallest number in dp[1] , this means that it can be a potential candidate to be included in dp[2].

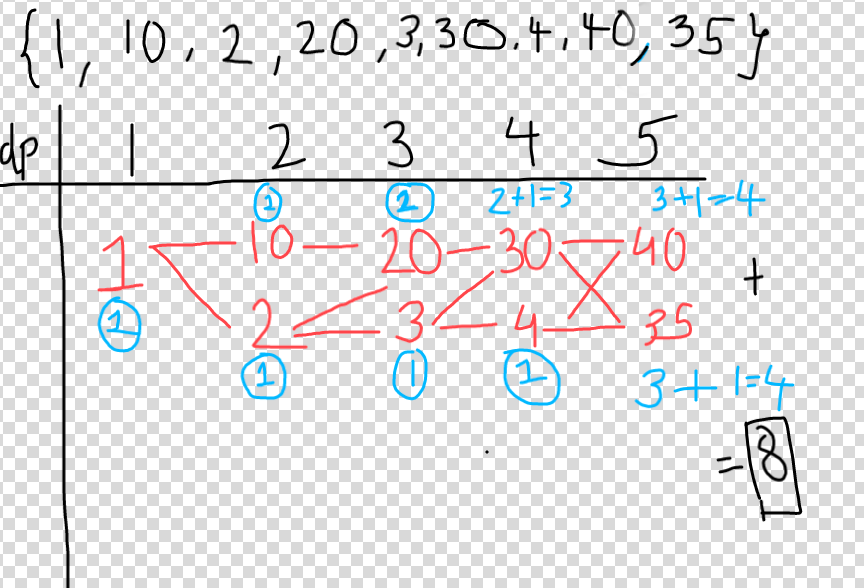
->Now if it’s also greater than smallest element in dp[2], but smaller(**or equal)** than smallest element in dp[3],means it will definitely be in dp[3] ,and **it will always be smaller than or equal to the previous smallest element in the group because if it would have been greater,we would have placed it in dp[4] or further to form a larger subsequence .**

->So we will place this new smallest element in in the end, so basically in every dp , elements are stored in non increasing order.

-> **So we can definitely binary search to see it falls in which dp row(means it can form what length subsequence at max).**

-> But one more important thing remains. How many elements in previous row are actually smaller than current element.(remember we compared it with smallest element in every vector, not all elements)

-> For that we maintain a **prefix sum.So if we imagine it as a tree and find out how many elements are smaller than it in previous vector,than basically we can extend an edge from all those smaller numbers to this number , so basically number of paths to all these numbers will also extend to this number .**

****

Just to understand prefix sum, see the last number 35, it’s greater than both 30 and 4 in previous dp row, so all paths from both 30 and 4 will extend to 35 which is 3+1=4,so instead of storind individual no of paths,means preSum[30]=3,and preSum[4]=1, store prefix sum, means preSum[30]=3,preSum[4]=3+1=4,which prefix sum to return we can return it by a formula easily.

Here 35 is greater than both so return preSum[4]=4 means count of last element(smallest element) as it will have prefix sum of whole vector, assume if instead of 35 , the number was 5, then we would have returned preSum[whole vector]-preSum[upto number 35 as 5<35] =**4-3=1.**

**As numbers would have been added in descending order only, prefix sum logic works.**

**Code :** [**https://leetcode.com/problems/number-of-longest-increasing-subsequence/submissions/**](https://leetcode.com/problems/number-of-longest-increasing-subsequence/submissions/)