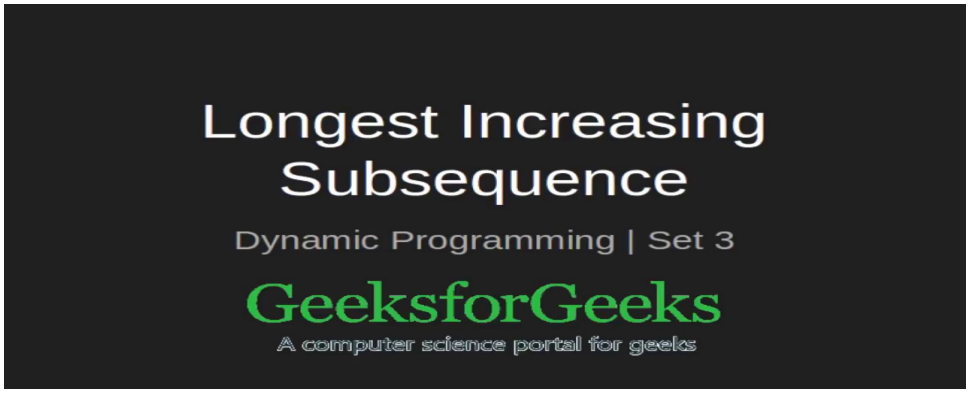
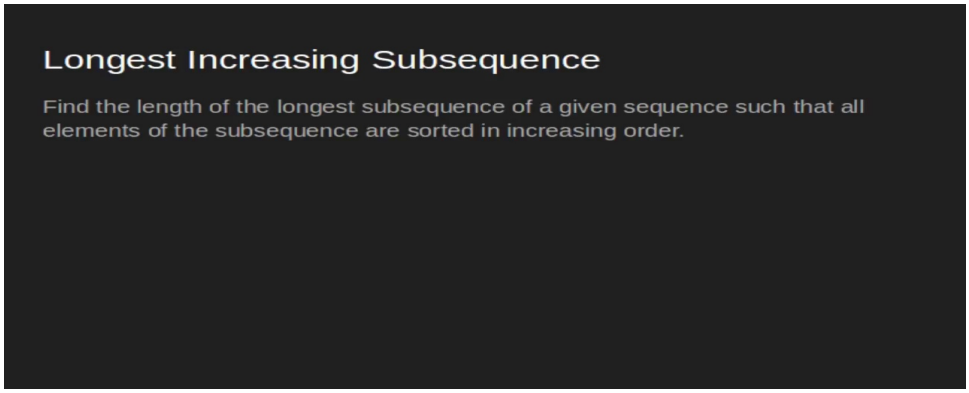
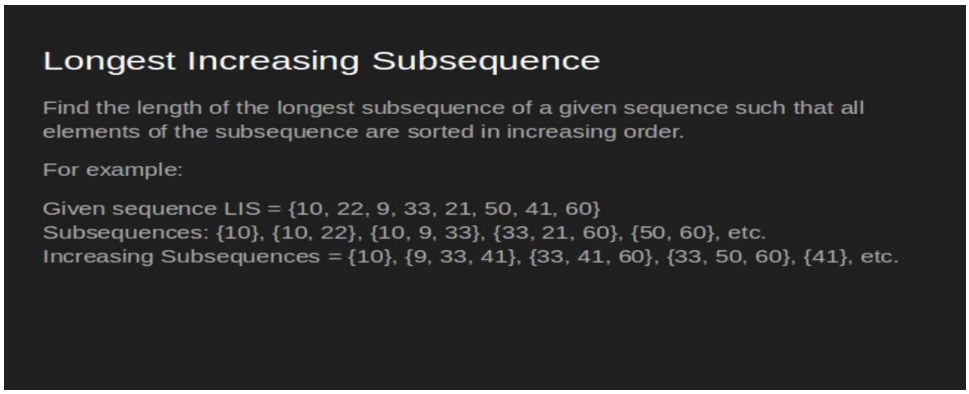


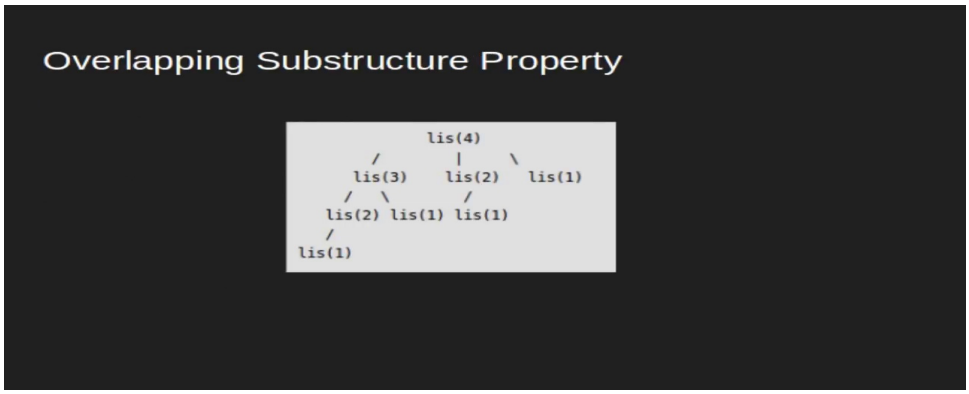
Today we will learn program for longest increasing subsequence.



Let us take an example.A single element is also a subsequence, for example 10.

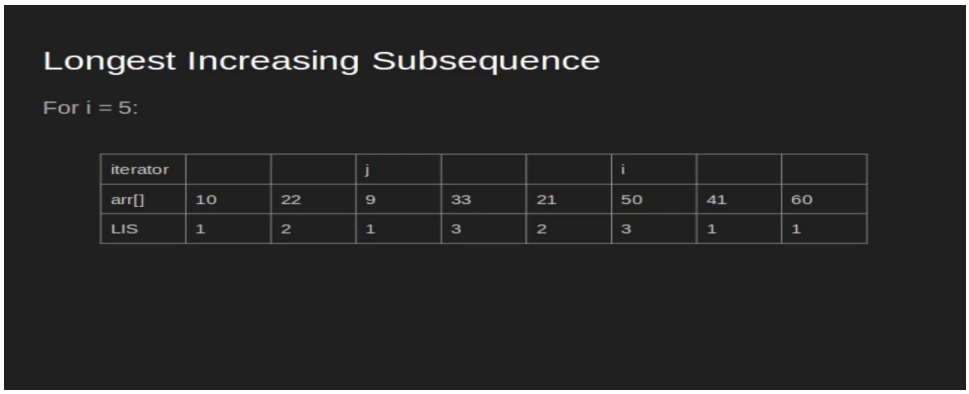


For the given example, we find that these are the two longest increasing subsequences 1020, two 3350 and 60.So our algorithm should return the length of list as five.

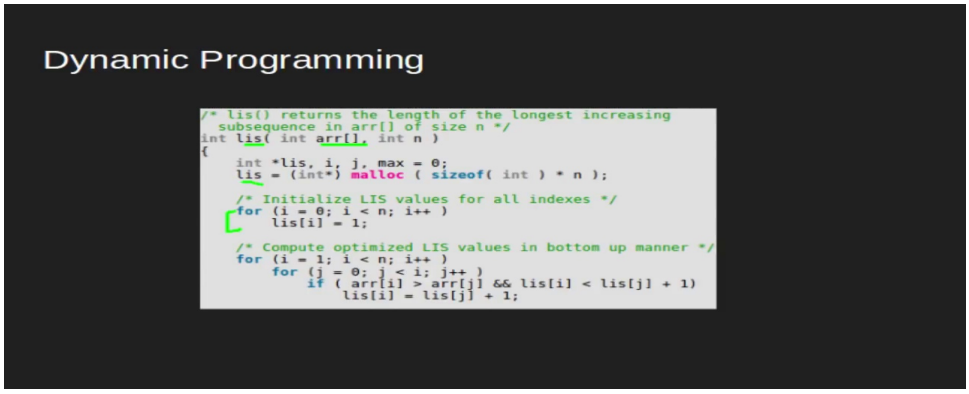


Subproblems list of two and lists of one are solved repeatedly.Liz is an error a that stores the length of longest increasing subsequence ending at the corresponding index of given array. Itself is an

increasing subsequence of length one.If array of I is greater than array of J, then we have an increasing subsequence.Note that for equal to 0 list of 0 is already one, so we initialize I with index one.For I equal to two is 9 greater than 10.Is 9 greater than 22? Yes so list of I becomes list of one that is 1 + 1 equal to two.Yes, so list of eye becomes list of J plus one that is 2 + 1 equal to three is 33 greater than nine. Yes, so listen I becomes list of one that is 112 is 21 greater than 22. Yes, Solis 05 becomes list of one that is 1 + 1 equal to two and maximum of two, two is 2, so we do not change value.Is 50 greater than 9? We need to make sure that we are updating maximum value of list but maximum of two, three is 3.



Is 50 greater than 33?It's 50 great than 21 yes. That is 2 + 1 equal to three maximum of four.Similarly, you can workout rest of the list are a final result will be this table.Now we can discuss dynamic programming code according to the logic we used earlier to fill the table.Area R consists of sequence and Liz is all located memory dynamically using malloc of same size as avarayr.We initialize list array with value 1.



If element at I is greater than element at J, we have found increasing subsequence.We then iterate over all the elements of list array and find maximum value.