

CS 313E (Mitra) Longest Increasing Subsequence Problem

Given an array of integers, your goal is to find the length of the longest increasing subsequence that is not necessarily contiguous. For example, given the array [3, 5, 1, 6, 2, 14, 51, 7, 12, 13, 8] the longest increasing subsequence is [3, 5, 6, 7, 12, 13] which has length 6.

This problem can be solved using dynamic programming where $lis[i]$ is the length of the longest increasing subsequence of the subarray $arr[0:i+1]$ using the following algorithm.

```
for k in range(0, i):  
    if arr[i] > arr[k]:  
        lis[i] = max(lis[i], lis[k] + 1)
```

Then, after we have all values in our LIS array, the answer is simply the max value in this array.

Given the example input array above, [3, 5, 1, 6, 2, 14, 51, 7, 12, 13], complete the table below.

Index, i	arr[i]	lis[i]	Longest Increasing Subsequence including arr[i] of arr[0:i+1]
0	3	1	[3]
1	5	2	[3, 5]
2	1	1	[1]
3	6	3	[3, 5, 6]
4	2		
5	14		
6	51		
7	7		
8	12		
9	13		
10	8		