300. Longest Increasing Subsequence

QuestionEditorial Solution
My Submissions

- Total Accepted: **40669**
- Total Submissions: **113569**
- Difficulty: **Medium**

Given an unsorted array of integers, find the length of longest increasing subsequence.

For example,

Given [10, 9, 2, 5, 3, 7, 101, 18],

The longest increasing subsequence is [2, 3, 7, 101], therefore the length is 4. Note that there may be more than one LIS combination, it is only necessary for you to return the length.

Your algorithm should run in O(n^2) complexity.

Follow up: Could you improve it to O(*n* log *n*) time complexity?

Credits:

Special thanks to @pbrother for adding this problem and creating all test cases.

Subscribe to see which companies asked this question

```
class Solution {
public:
    int lengthOfLIS(vector<int>& nums) {
        if(nums.size() == 0) return 0;
        if(nums.size() == 1) return 1;
        vector<int> dist(nums.size(),1);
        int max = INT_MIN;
        for(int i=1;i<nums.size();i++)
        {
            for(int j=0;j<i;j++)
            {
                if(nums[i]>nums[j] && dist[j]+1>dist[i])
                    dist[i] = dist[j] + 1;
            }
            if(dist[i]>max)
                max = dist[i];
        }
        return max;
    }
};
```