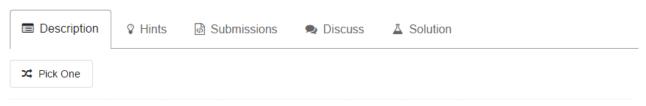
300. Longest Increasing Subsequence



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

Example:

```
Input: [10,9,2,5,3,7,101,18]
Output: 4
Explanation: The longest increasing subsequence is [2,3,7,101], therefore the length is 4.
```

Note:

- . There may be more than one LIS combination, it is only necessary for you to return the length.
- Your algorithm should run in O(n²) complexity.

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

```
public class L300 {

public int lengthOfLIS(int[] nums) {

TreeSet<Integer> set = new TreeSet<>();
//为什么treeset可以,因为treeset是排好序了的,剔除一个增加一个,不会影响最长的个数 for(int num : nums) {

/*

* ceiling是查询set中,返回的是比这个数大的最小值

* 如果没有大过它的或者是无法比较,则返回null

*/

Integer ceil = set.ceiling(num);
    if(ceil != null)
        set.remove(ceil);
    set.add(num);
}

return set.size();
}
```