300. Longest Increasing Subsequence

题目描述：https://leetcode.com/problems/longest-increasing-subsequence/

> 判断一个无序数组的最长上升子序列

解题思路：

1. dp f[i] = 以i结尾的最长上升子序列
   f[i] = max{ f[k] + 1, if n[k] < n[i] }
2. 二分

代码DP：

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

代码BS：

```
int lengthOfLIS(vector<int>& nums) {
    vector<int> res;
    for(int i=0; i<nums.size(); i++) {
        auto it = std::lower_bound(res.begin(), res.end(), nums[i]);
        if(it==res.end()) res.push_back(nums[i]);
        else *it = nums[i];
    }
    return res.size();
}
```