1. 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?

**解** 动态规划[https://www.cnblogs.com/vinnson/p/10845125.html]

class Solution {  
public:  
 int lengthOfLIS(vector<int>& nums) {  
 int n = nums.size();  
 if(n == 0)return 0;  
 int \*dp = new int[n];  
 for(int i = 0; i < n; ++i)dp[i] = 1;  
 int ans = 1;  
 for(int i = 1; i < n; ++i){  
 for(int j = 0; j < i; ++j){  
 if(nums[i] > nums[j])dp[i] = max(dp[i], dp[j]+1);  
 }  
 ans = max(ans, dp[i]);  
 }  
 delete []dp;  
 return ans;  
 }  
};