一、题目说明

题目300. Longest Increasing Subsequence，给一列无序的整数，找出最大递增序列的长度。难度是Medium！

二、我的解答

这个题目用dp解决，开始想简单了。其中`dp[i]`表示，前面比`nums[i]`小的数量且递增的个数。

```
class Solution{
    public:
        int lengthOfLIS(vector<int>& nums){
            int len = nums.size();
            if(len<1) return 0;

            else if(len<2) return 1;

            vector<int> dp(len+1,0);
            int num = 1;
            //比当前数字小的个数
            dp[0] = 1;
            bool hasLess = false;
            for(int i=1;i<nums.size();i++){
                if(nums[i]>nums[i-1]){
                    dp[i] = dp[i-1]+1;
                    for(int j=i-2;j>=0;j--){
                        if(nums[i]>nums[j]){
                            if(dp[j]+1>dp[i]) dp[i] = dp[j]+1;
                        }
                    }
                }else{
                    hasLess = false;
                    for(int j=i-2;j>=0;j--){
                        if(nums[i]>nums[j]){
                            dp[i] = dp[j]+1;
                            hasLess = true;
                            break;
                        }
                    }
                    if(! hasLess )dp[i] = 1;
                }
                if(dp[i]>num) num = dp[i];
            }

            for(int i=0;i<nums.size();i++){
                cout<<dp[i]<<"->";
            }
            return num;
        }
};
```

性能如下：

```
Runtime: 32 ms, faster than 68.27% of C++ online submissions for Longest
Increasing Subsequence.
Memory Usage: 8.8 MB, less than 51.56% of C++ online submissions for Longest
Increasing Subsequence.
```

三、优化措施