1. Insert Interval

Hard

1090136Share

Given a set of *non-overlapping* intervals, insert a new interval into the intervals (merge if necessary).

You may assume that the intervals were initially sorted according to their start times.

**Example 1:**

Input: intervals = [[1,3],[6,9]], newInterval = [2,5]  
Output: [[1,5],[6,9]]

**Example 2:**

Input: intervals = [[1,2],[3,5],[6,7],[8,10],[12,16]], newInterval = [4,8]  
Output: [[1,2],[3,10],[12,16]]  
Explanation: Because the new interval [4,8] overlaps with [3,5],[6,7],[8,10].

**解**

解法1 直接扫描，时间复杂度为O(n)

解法2 二分查找插入区间所在的位置

在单调递增数列中查找：

* 第一个大于x的位置，若A[mid] > x，则pos 在[l, mid]
* 第一个大于或者等于x的位置, 若A[mid] >= x，则pos 在[l, mid]
* 最后一个小于x的位置, 若A[mid] < x，则pos 在[mid, r]
* 最后一个小于或者等于x的位置，若A[mid] <= x，则pos 在[mid, r]

class Solution {  
public:  
 vector<vector<int>> insert(vector<vector<int>>& intervals, vector<int>& newInterval) {  
 if(intervals.size() == 0)return vector<vector<int>>({newInterval});  
 int left = 0, right = intervals.size(), mid;  
 //找第一个大于或等于x的位置  
 while(left < right){  
 mid = (left + right) / 2;  
 //printf("mid: %d\n", intervals[mid][1]);  
 if(intervals[mid][1] >= newInterval[0])right = mid;//x在[l, mid - 1]  
 else left = mid + 1;  
 //printf("[left, right] : [%d, %d]\n", left, right);  
 }  
 int pos1 = left;  
 left = pos1;  
 right = intervals.size();  
 //找第一个大于x的位置  
 while(left < right){  
 mid = (left + right) / 2;  
 //printf("mid: %d\n", intervals[mid][0]);  
 if(intervals[mid][0] > newInterval[1])right = mid;  
 else left = mid + 1;  
 //printf("[left, right] : [%d, %d]\n", left, right);  
 }  
 int pos2 = left - 1;  
 //printf("pos1 : %d pos2 : %d\n", pos1, pos2);  
 vector<vector<int>>ans;  
 for(int i = 0; i < pos1; ++i)ans.push\_back(intervals[i]);  
 ans.push\_back({pos1 == intervals.size() ? newInterval[0] : min(newInterval[0], intervals[pos1][0]), pos2 == -1 ? newInterval[1] : max(newInterval[1], intervals[pos2][1])});  
 for(int i = pos2 + 1; i < intervals.size(); ++i)ans.push\_back(intervals[i]);  
 return ans;  
 }  
};