第六周练习

张睿恒 2024302182001

完成题目：LeetCode215 简单

LeetCode347 中等

Leetcode23 中等

LeetCode480 中等

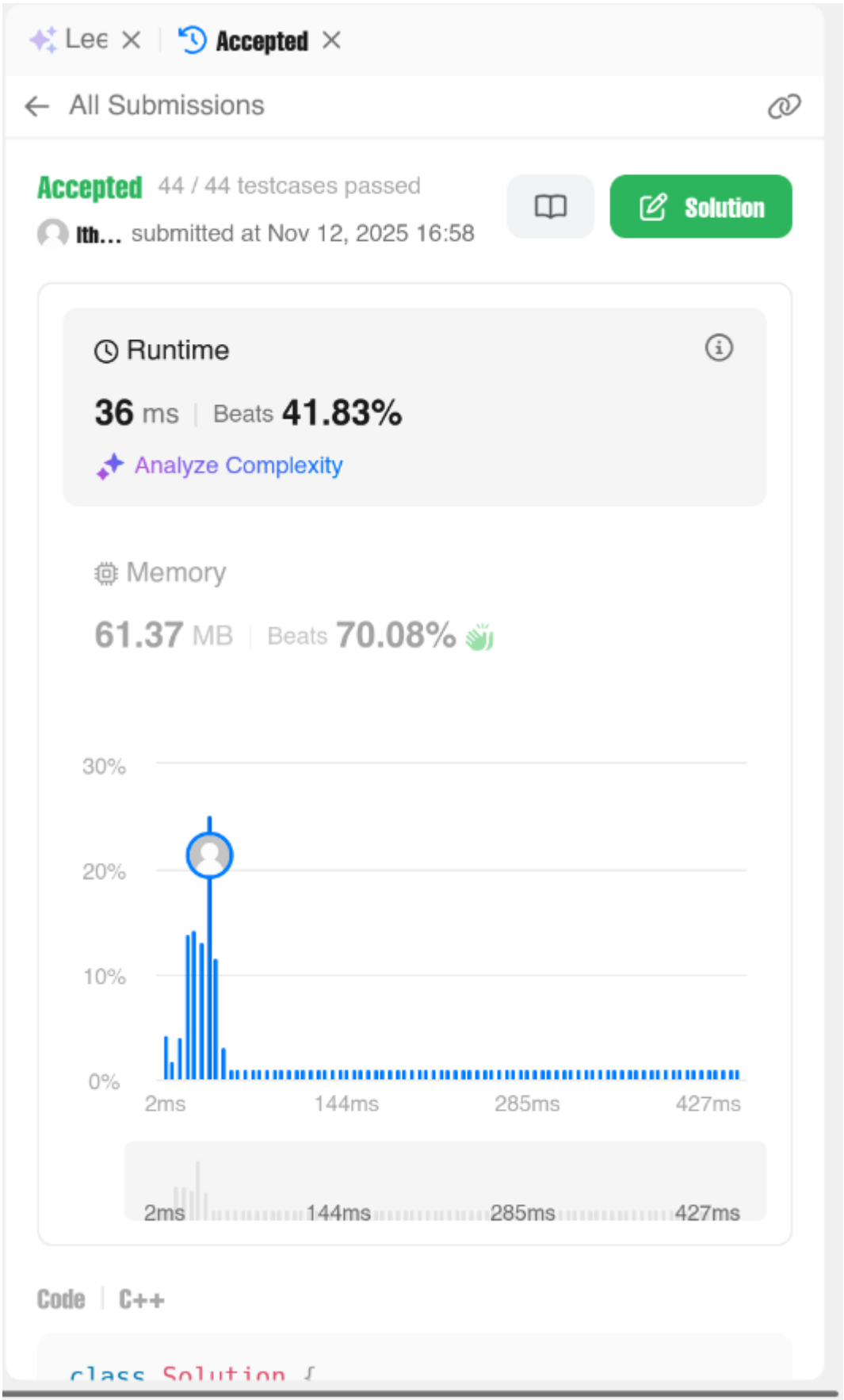
LeetCode295 困难

LeetCode218 困难

T1 LeetCode215

```
class Solution {
public:
    int findKthLargest(vector<int>& nums, int k) {
        priority_queue <int,vector <int>,greater <> > que;
        for(int i=0;i<nums.size();++i){
            que.push(nums[i]);
            if(que.size() > k) que.pop();
        }

        return que.top();
    }
};
```



T2 LeetCode347

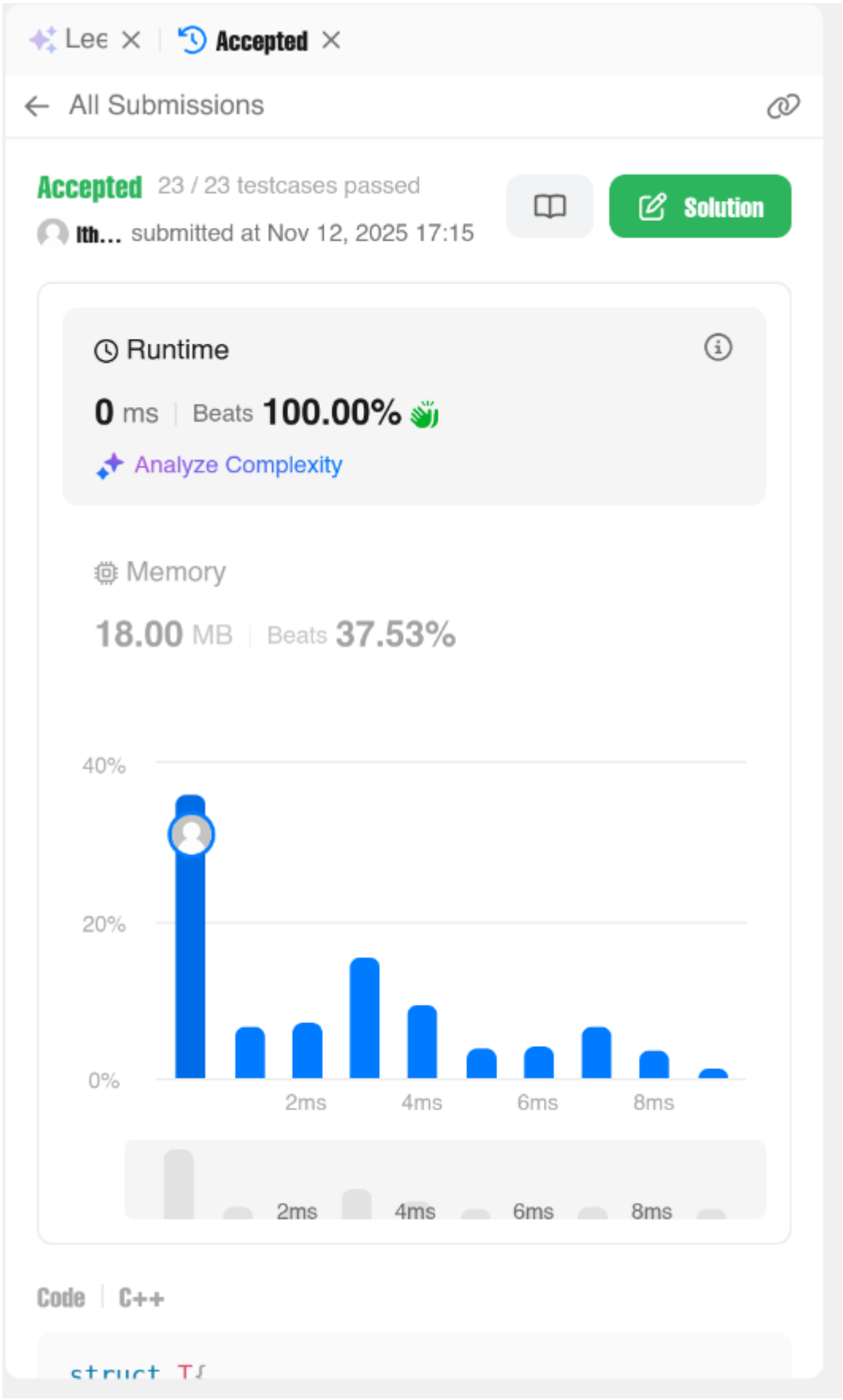
```
struct T{
    int num,freq;
};


class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        int n = nums.size();
        vector <int> ans;
        unordered_map <int,int> cnt;
        auto cmp = [](const T& a,const T& b){return a.freq > b.freq;};
        priority_queue <T, vector <T>, decltype(cmp)> que(cmp);

        for(int i=0;i<n;++i){
            cnt[nums[i]]++;
        }
        for (const auto& [num, freq] : cnt) {
            que.emplace(num, freq);
            if (que.size() > k) que.pop();
        }

        while (!que.empty())
            ans.push_back(que.top().num), que.pop();

        return ans;
    }
};
```



T3 LeetCode23

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
struct nodeCmp{
    bool operator()(const ListNode* a,const ListNode* b) const{
        return a->val > b->val;
    }
};

class Solution {
public:
    ListNode* mergeKLists(vector<ListNode*>& lists) {
        ListNode ans;
        ListNode* now;
        now = &ans;
        priority_queue <ListNode*, vector <ListNode*>, nodeCmp> que;

        for(ListNode* listnode : lists){
            if(listnode != NULL) que.push(listnode);
        }

        while(!que.empty()){
            ListNode* min1 = que.top();
            que.pop();
            if(min1->next != NULL) que.push(min1->next);
            now->next = min1;
            now = now->next;
        }

        return ans.next;
    }
};
```

Lee × | Accepted ×

← All Submissions

Accepted 134 / 134 testcases passed

Itheng submitted at Nov 12, 2025 18:12

Editorial | Solution

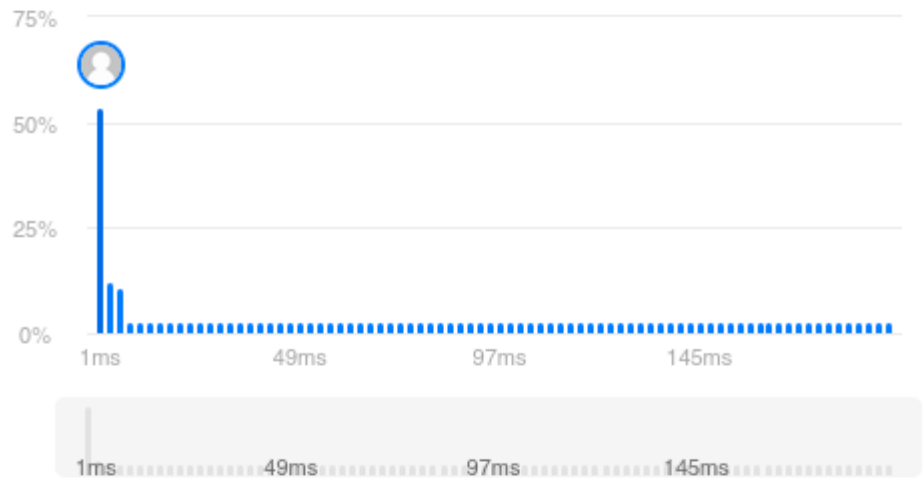
Runtime

0 ms | Beats **100.00%**

Analyze Complexity

Memory

18.32 MB | Beats **81.26%**



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next)
```

View more

T4 Leetcode480

```
class Solution {
public:
    vector<double> medianSlidingWindow(vector<int>& nums, int k) {
    vector<double> ans;
    multiset<double> window(nums.begin(), nums.begin() + k);
    auto it = next(window.begin(), (k - 1) / 2);

    for (int i = k;; ++i) {
      const double median = k % 2 == 0 ? (*it + *next(it)) / 2.0 : *it;
      ans.push_back(median);
      if (i == nums.size()) break;
      window.insert(nums[i]);
      if (nums[i] < *it) --it;
      if (nums[i - k] <= *it) ++it;
      window.erase(window.lower_bound(nums[i - k]));
    }

    return ans;
    }
};
```

Lee × | Accepted ×

← All Submissions

Accepted 44 / 44 testcases passed

Itheng submitted at Nov 12, 2025 18:24

Editorial | Solution

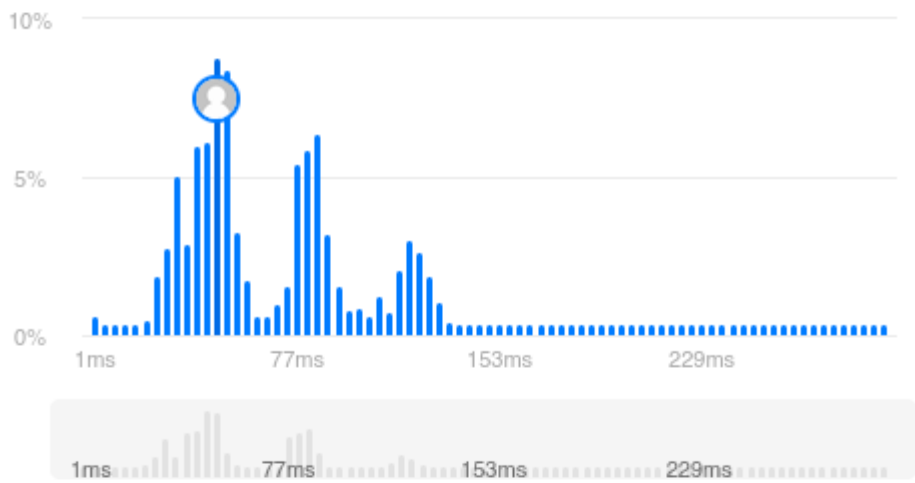
Runtime

47 ms | Beats **73.53%**

Analyze Complexity

Memory

43.19 MB | Beats **49.92%**



Code | C++

```
class Solution {
public:
    vector<double> medianSlidingWindow(vector<int>& nums, :
    vector<double> ans;
    multiset<double> window(nums.begin(), nums.begin() + k
    auto it = next(window.begin(), (k - 1) / 2);

    for (int i = k;; ++i) {
```

View more

T5 LeetCode295

```
class MedianFinder {
public:
    MedianFinder() {

    }

    void addNum(int num) {
        if(queMax.empty() or num <= queMax.top()) queMax.push(num);
        else queMin.push(num);

        if(queMax.size() < queMin.size()){
            queMax.push(queMin.top());
            queMin.pop();
        }
        else if(queMax.size()-queMin.size() > 1){
            queMin.push(queMax.top());
            queMax.pop();
        }
    }

    double findMedian() {
        if(queMax.size() == queMin.size()) return (queMax.top()+queMin.top())/2.0;
        return queMax.top();
    }
private:
    priority_queue <int> queMax;
    priority_queue <int, vector <int>, greater<int> > queMin;
};

/**
 * Your MedianFinder object will be instantiated and called as such:
 * MedianFinder* obj = new MedianFinder();
 * obj->addNum(num);
 * double param_2 = obj->findMedian();
 */
```

Lee × | Accepted ×

← All Submissions

Accepted 22 / 22 testcases passed

Itheng submitted at Nov 12, 2025 18:43

Editorial | Solution

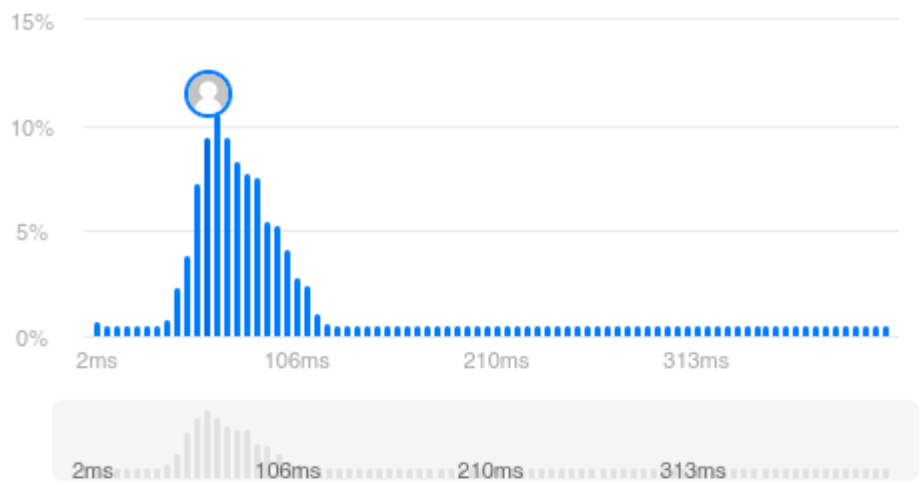
Runtime

63 ms | Beats **77.20%**

Analyze Complexity

Memory

148.57 MB | Beats **61.39%**



Code | C++

```
class MedianFinder {
public:
    MedianFinder() {

    }

    void addNum(int num) {
        if(queMax.empty() or num <= queMax.top()) queMax.pu
```

View more

T6 LeetCode218

```
class Solution {
 public:
  vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
    vector<vector<int>> ans;
    vector<vector<int>> events;  // [(Li, Hi) | (Ri, -Hi)]

    for (const vector<int>& b : buildings) {
      events.push_back({b[0], b[2]});
      events.push_back({b[1], -b[2]});  // Minus means leaving.
    }

    ranges::sort(events, ranges::less{}, [](const vector<int>& event) {
      return pair<int, int>{event[0], -event[1]};
    });

    for (const vector<int>& event : events) {
      const int x = event[0];
      const int h = abs(event[1]);
      const int isEntering = event[1] > 0;
      if (isEntering) {
        if (h > maxHeight()) ans.push_back({x, h});
        set.insert(h);
      }
      else {
        set.erase(set.equal_range(h).first);
        if (h > maxHeight()) ans.push_back({x, maxHeight()});
      }
    }

    return ans;
  }

 private:
  multiset<int> set;

  int maxHeight() const {
    return set.empty() ? 0 : *set.rbegin();
  }
};
```

Lee × | Accepted ×

← All Submissions

Accepted 44 / 44 testcases passed

Itheng submitted at Nov 12, 2025 18:55

Editorial | Solution

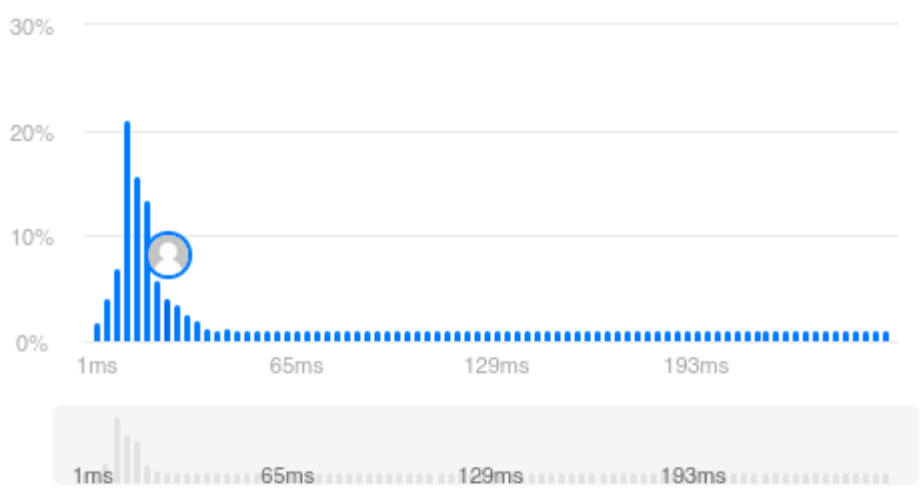
Runtime

25 ms | Beats **29.05%**

Analyze Complexity

Memory

31.68 MB | Beats **25.45%**



Code | C++

```cpp
class Solution {
 public:
  vector<vector<int>> getSkyline(vector<vector<int>>& build
    vector<vector<int>> ans;
    vector<vector<int>> events;  // [(Li, Hi) | (Ri, -Hi)]

    for (const vector<int>& b : buildings) {
      events.push_back({b[0], b[2]});
```

View more