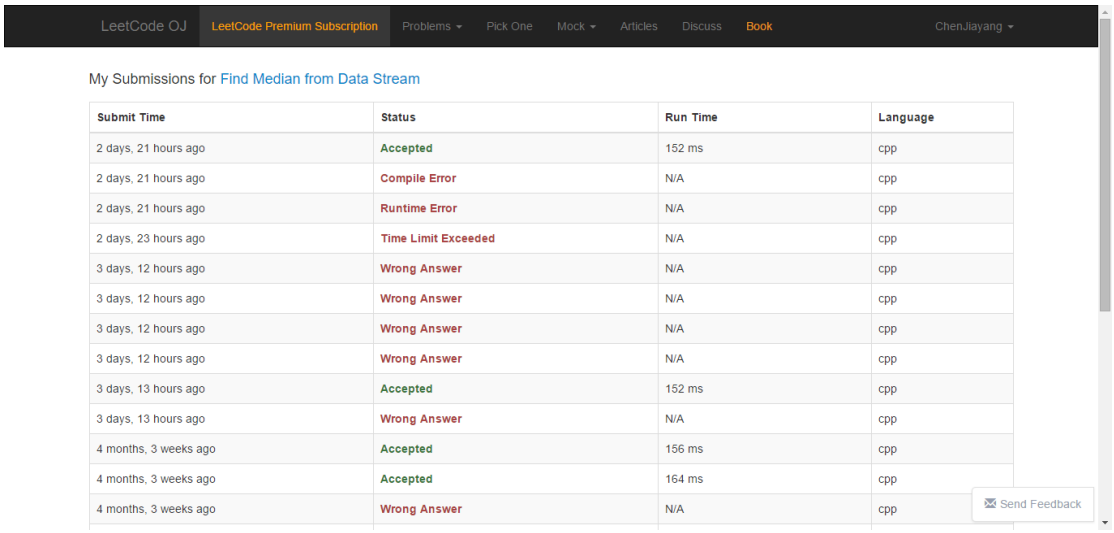
PA 4

LeetCode ID：ChenJiayang

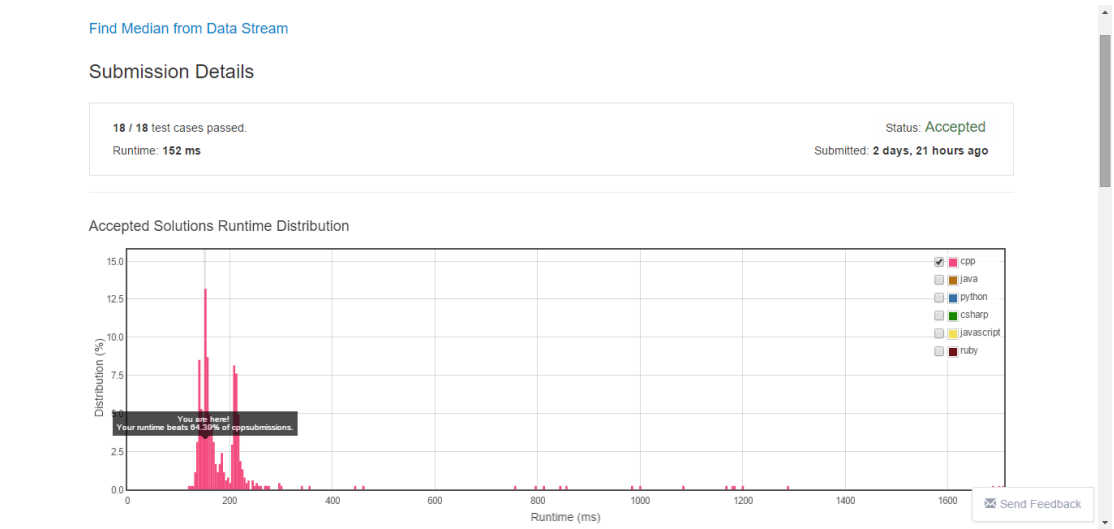


LeetCode OJ | LeetCode Premium Subscription | Problems | Pick One | Mock | Articles | Discuss | Book | ChenJiayang

My Submissions for Find Median from Data Stream

Submit Time	Status	Run Time	Language
2 days, 21 hours ago	Accepted	152 ms	cpp
2 days, 21 hours ago	Compile Error	N/A	cpp
2 days, 21 hours ago	Runtime Error	N/A	cpp
2 days, 23 hours ago	Time Limit Exceeded	N/A	cpp
3 days, 12 hours ago	Wrong Answer	N/A	cpp
3 days, 12 hours ago	Wrong Answer	N/A	cpp
3 days, 12 hours ago	Wrong Answer	N/A	cpp
3 days, 12 hours ago	Wrong Answer	N/A	cpp
3 days, 13 hours ago	Accepted	152 ms	cpp
3 days, 13 hours ago	Wrong Answer	N/A	cpp
4 months, 3 weeks ago	Accepted	156 ms	cpp
4 months, 3 weeks ago	Accepted	164 ms	cpp
4 months, 3 weeks ago	Wrong Answer	N/A	cpp

Send Feedback



Find Median from Data Stream

Submission Details

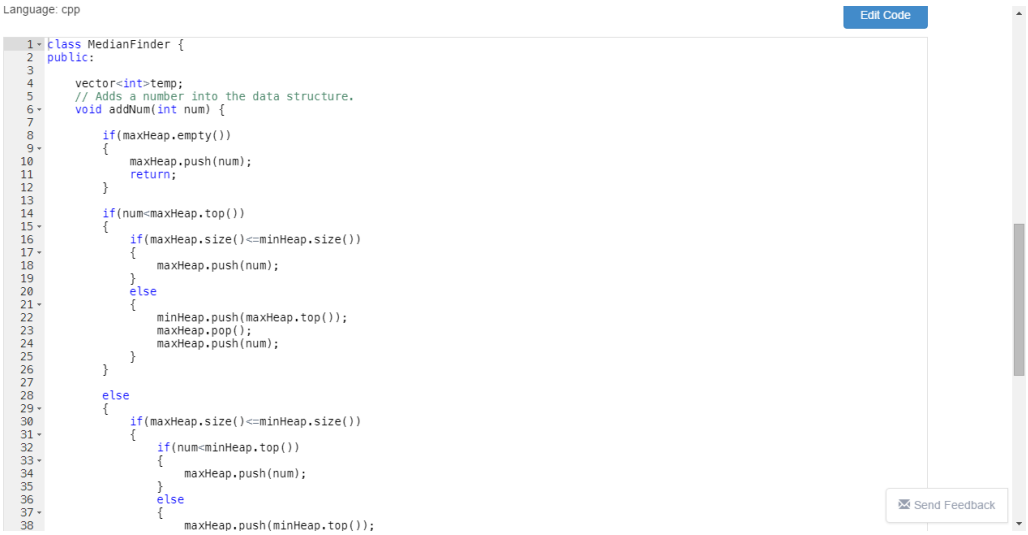
18 / 18 test cases passed.
Runtime: **152 ms**

Status: Accepted
Submitted: **2 days, 21 hours ago**

Accepted Solutions Runtime Distribution

Send Feedback

Code (C++)

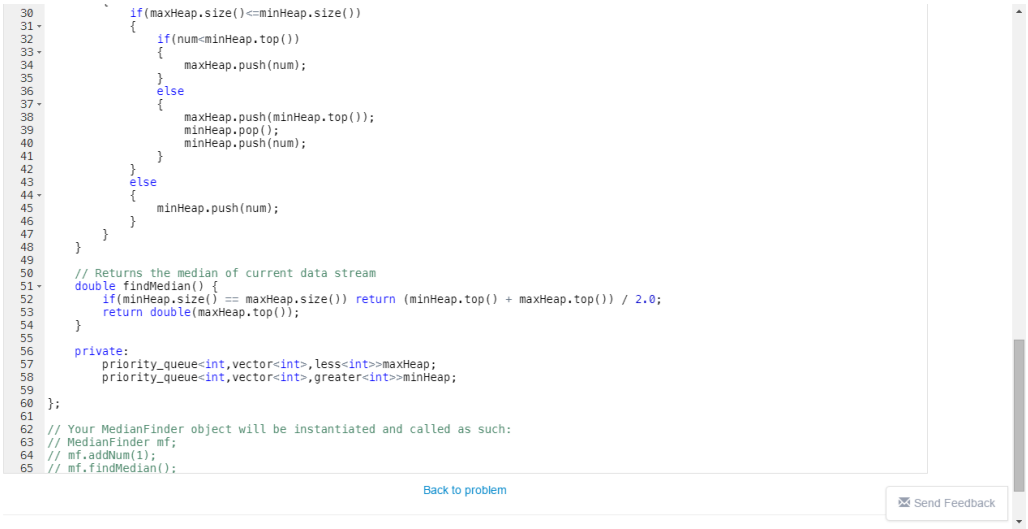


Language: cpp

Edit Code

```cpp
class MedianFinder {
public:

    vector<int>temp;
    // Adds a number into the data structure.
    void addNum(int num) {

        if(maxHeap.empty())
        {
            maxHeap.push(num);
            return;
        }

        if(num<maxHeap.top())
        {
            if(maxHeap.size()<=minHeap.size())
            {
                maxHeap.push(num);
            }
            else
            {
                minHeap.push(maxHeap.top());
                maxHeap.pop();
                maxHeap.push(num);
            }
        }

        else
        {
            if(maxHeap.size()<=minHeap.size())
            {
                if(num<minHeap.top())
                {
                    maxHeap.push(num);
                }
                else
                {
                    maxHeap.push(minHeap.top());
```

Send Feedback



```
        `   if(maxHeap.size()<=minHeap.size())
            {
                if(num<minHeap.top())
                {
                    maxHeap.push(num);
                }
                else
                {
                    maxHeap.push(minHeap.top());
                    minHeap.pop();
                    minHeap.push(num);
                }
            }
            else
            {
                minHeap.push(num);
            }
        }
    }

    // Returns the median of current data stream
    double findMedian() {
        if(minHeap.size() == maxHeap.size()) return (minHeap.top() + maxHeap.top()) / 2.0;
        return double(maxHeap.top());
    }

    private:
        priority_queue<int,vector<int>,less<int>>maxHeap;
        priority_queue<int,vector<int>,greater<int>>minHeap;

};

// Your MedianFinder object will be instantiated and called as such:
// MedianFinder mf;
// mf.addNum(1);
// mf.findMedian();
```


关于此题更详细的解答请见：

http://chenjiayang.me/2016/03/12/leetcode-295%E9%A2%98%E8%A7%A3/