

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

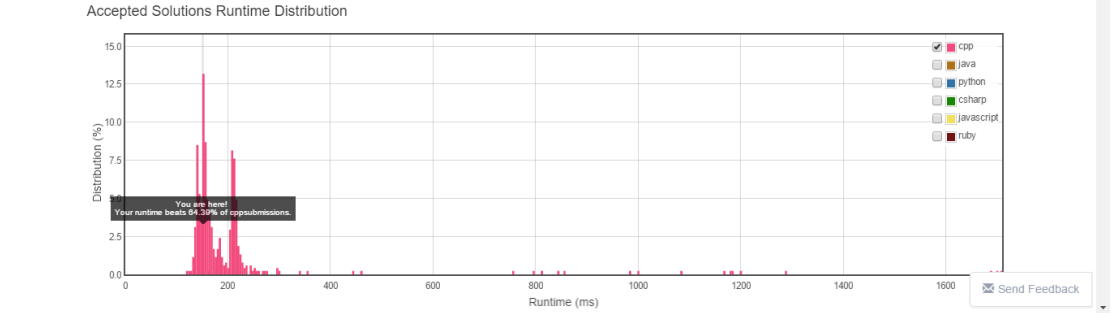
Send Feedback

[Find Median from Data Stream](#)

Submission Details

18 / 18 test cases passed.
Runtime: 152 ms

Status: Accepted
Submitted: 2 days, 21 hours ago



Code (C++)

Language: cpp **Edit Code**

```
1 class MedianFinder {
2 public:
3
4     vector<int>temp;
5     // Adds a number into the data structure.
6     void addNum(int num) {
7
8         if(maxHeap.empty())
9         {
10             maxHeap.push(num);
11             return;
12         }
13
14         if(num<maxHeap.top())
15         {
16             if(maxHeap.size()<=minHeap.size())
17             {
18                 maxHeap.push(num);
19             }
20             else
21             {
22                 minHeap.push(maxHeap.top());
23                 maxHeap.pop();
24                 maxHeap.push(num);
25             }
26         }
27         else
28         {
29             if(maxHeap.size()<=minHeap.size())
30             {
31                 if(num<minHeap.top())
32                 {
33                     maxHeap.push(num);
34                 }
35                 else
36                 {
37                     maxHeap.push(minHeap.top());
38                 }
39             }
40             else
41             {
42                 minHeap.push(num);
43             }
44         }
45     }
46
47     double findMedian() {
48         if(temp.size()%2==0)
49         {
50             return (maxHeap.top()+minHeap.top())/2.0;
51         }
52         else
53         {
54             return minHeap.top();
55         }
56     }
57
58 }
```

Send Feedback

```

30         if(maxHeap.size() <= minHeap.size())
31         {
32             if(num < minHeap.top())
33             {
34                 maxHeap.push(num);
35             }
36             else
37             {
38                 maxHeap.push(minHeap.top());
39                 minHeap.pop();
40                 minHeap.push(num);
41             }
42         }
43         else
44         {
45             minHeap.push(num);
46         }
47     }
48 }
49
50 // Returns the median of current data stream
51 double findMedian() {
52     if(minHeap.size() == maxHeap.size()) return (minHeap.top() + maxHeap.top()) / 2.0;
53     return double(maxHeap.top());
54 }
55
56 private:
57     priority_queue<int, vector<int>, less<int>> maxHeap;
58     priority_queue<int, vector<int>, greater<int>> minHeap;
59 };
60
61 // Your MedianFinder object will be instantiated and called as such:
62 // MedianFinder mf;
63 // mf.addNum(1);
64 // mf.findMedian();

```

[Back to problem](#)

 Send Feedback

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/>