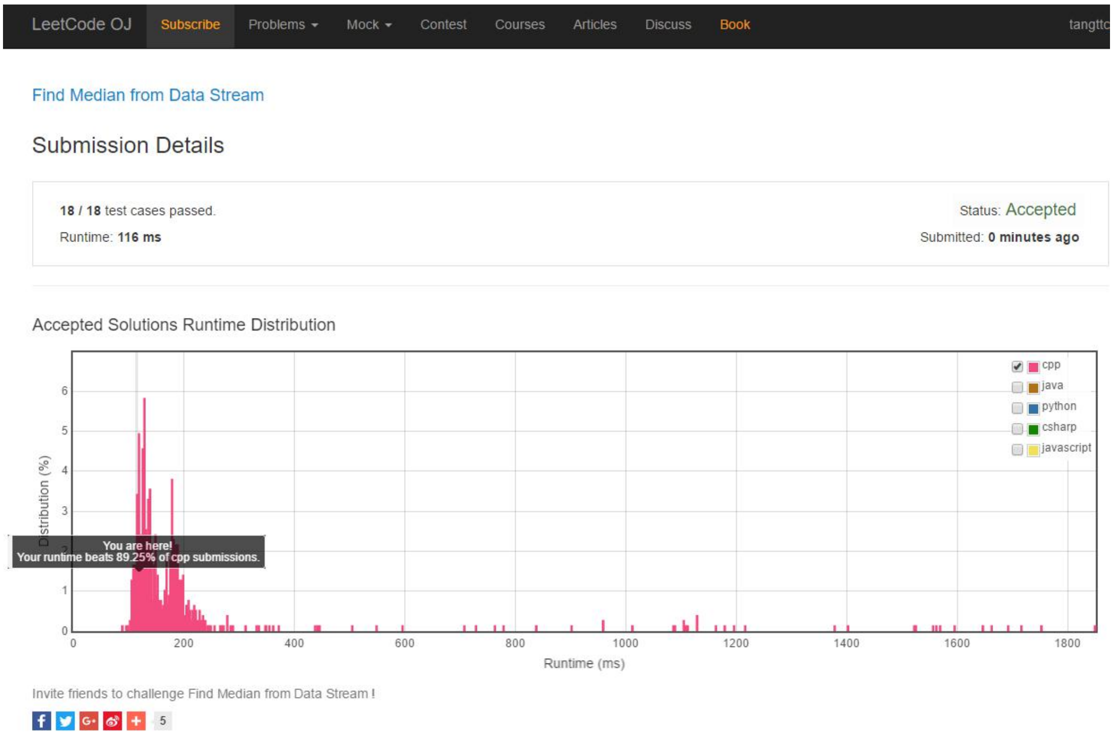
PA3 唐天成 515030910287

Leetcode 295:Find Median from Data Stream



Find Median from Data Stream

Submission Details

18 / 18 test cases passed.

Runtime: 116 ms

Status: Accepted

Submitted: 0 minutes ago

Accepted Solutions Runtime Distribution

Invite friends to challenge Find Median from Data Stream !

Code(C++):

```
#include <iostream>
#include <queue>
using namespace std;

class MedianFinder {
public:
    struct cmp{
        bool operator()(int &a, int &b){
            return a > b;
        }
    };
    // Adds a number into the data structure.
    void addNum(int num) {
        if (left_size!=0 ){ // first push the num
            if (num <= left.top()){
                left.push(num);
                left_size++;
            }
            else{
                right.push(num);
                right_size++;
            }
        }
    else
    {
        if (right_size == 0){
            left.push(num);
            left_size++;
        }
        else {
            if (num >= right.top()){
                right.push(num);
                right_size++;
            }
            else{
                left.push(num);
                left_size++;
            }
        }

    }
    if (left_size <= (right_size - 2)){ // then maintain the size of the two priority queues
        left.push(right.top());
        right.pop();
        left_size++;
        right_size--;
    }
    else if (left_size >= (right_size + 2)){
        right.push(left.top());
        left.pop();
        left_size--;
        right_size++;
    }
}

// Returns the median of current data stream
double findMedian() {
    if (left_size == right_size){
        return (left.top()+right.top())*0.5;
    }
    else if (left_size < right_size){
        return right.top();
    }
    else{
        return left.top();
    }
}
```

```
private:
        priority_queue<int> left; // the bigest is on the top
        priority_queue<int, vector<int>, cmp > right; // the smallest is on the top
        int left_size = 0;
        int right_size = 0;
};

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