Find Median from a Data Stream

Try to solve the Find Median from a Data Stream problem.

We'll cover the following

Statement

Implement a data structure that'll store a dynamically growing list of integers and provide access to their median in $O(1)$.

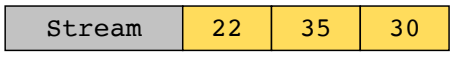
Constraints:

- $-10^5 \leq$ `num` $\leq 10^5$, where `num` is an integer received from the data stream.
- There will be at least one element in the data structure before the median is computed.
- At most, 5×10^4 calls will be made to the function that calculates the median.

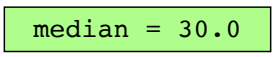
Examples

Sample example 1

Input



Output



1 of 2

Understand the problem

Take a moment to make sure you've correctly understood the problem. The quiz below will help you check if you're solving the correct problem:

Find Median from a Data Stream

1 What should be the output if the following number stream is given as input?

[12, 14, 36, 54]

A) 25.0

B) 29.0

C) 36.0

D) 14.0

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Split up incoming numbers into two lists—small and large. Those that are smaller than the current middle element are small, and those that are larger than it are large.

Maintain these lists as heaps so that the root of the small heap is the largest element in it and the root of large heap is the smallest element in it.

If the size of the large heap is greater than the size of small heap or, if size of small heap is greater than the size of the large heap + 1, rebalance the heaps.

If the number of elements is even, the median is the mean of the root of the two heaps.

Else, it's the root of the small heap.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in the following coding playground:

Java

usercode > MedianOfStream.java

```java
import java.util.*;

class MedianOfAStream {
  public MedianOfAStream() {

  }

  public void insertNum(int num) {
    // Write your code here
    return;
  }

  public double findMedian() {
      // Write your code here
      return 0.0;
  }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

["MedianOfStream","insertNum","insertNum","findMedian","insertNum","findMedian"]

Input #2

[[],[1],[2],[],[3],[]]

Find Median from a Data Stream

Back

Solution: Maximize Ca...

Next

Solution: Find Median ...

Mark as Completed