Our Solution(s) Run Code

Solution 1

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class ContinuousMedianHandler {
  constructor() {
    this.lowers = new Heap(MAX_HEAP_FUNC, []);
    this.greaters = new Heap(MIN_HEAP_FUNC, []);
    this.median = null;
  }

  // O(log(n)) time | O(n) space
  insert(number) {
    if (!this.lowers.length || number < this.lowers.peek()) {
      this.lowers.insert(number);
    } else {
      this.greaters.insert(number);
    }
    this.rebalanceHeaps();
    this.updateMedian();
  }

  rebalanceHeaps() {
    if (this.lowers.length - this.greaters.length === 2) {
      this.greaters.insert(this.lowers.remove());
    } else if (this.greaters.length - this.lowers.length === 2) {
      this.lowers.insert(this.greaters.remove());
    }
  }

  updateMedian() {
    if (this.lowers.length === this.greaters.length) {
      this.median = (this.lowers.peek() + this.greaters.peek()) / 2;
    } else if (this.lowers.length > this.greaters.length) {
      this.median = this.lowers.peek();
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
// Do not edit the class below except for
// the insert method. Feel free to add new
// properties and methods to the class.
class ContinuousMedianHandler {
  constructor(value) {
    // Write your code here.
    this.median = null;
  }

  insert(number) {
    // Write your code here.
  }

  getMedian() {
    return this.median;
  }
}

// Do not edit the line below.
exports.ContinuousMedianHandler = ContinuousMedianHandler;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.