

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System;
4 using System.Collections.Generic;
5
6 public class Program {
7     public static bool MAX_HEAP_FUNC(int a, int b) {
8         return a > b;
9     }
10
11     public static bool MIN_HEAP_FUNC(int a, int b) {
12         return a < b;
13     }
14
15     public class ContinuousMedianHandler {
16         public Heap lowers;
17         public Heap greater;
18         public double median = 0;
19
20         public ContinuousMedianHandler() {
21             this.lower = new Heap(Program.MAX_HEAP_FUNC, new List<int>());
22             this.greater = new Heap(Program.MIN_HEAP_FUNC, new List<int>());
23             this.median = 0;
24         }
25
26         // O(log(n)) time | O(n) space
27         public void Insert(int number) {
28             if (lowers.length == 0 || number < lowers.peek()) {
29                 this.lower.Insert(number);
30             } else {
31                 this.greater.Insert(number);
32             }
33             this.rebalanceHeaps();
34         }
35     }
36 }
```

Solution 1 Solution 2 Solution 3

```
1 // Do not edit the class below except for
2 // the Insert method. Feel free to add new
3 // properties and methods to the class.
4 public class Program {
5     public class ContinuousMedianHandler {
6         public double median = 0;
7
8         public void Insert(int number) {
9             // Write your code here.
10        }
11
12        public double GetMedian() {
13            return median;
14        }
15    }
16 }
17
```

Our Tests

```
1 using System.Collections.Generic;
2
3 public class Program {
4     public class ContinuousMedianHandler {
5         public Heap lowers;
6         public Heap greater;
7         public double median = 0;
8
9         public ContinuousMedianHandler() {
10             this.lower = new Heap(MAX_HEAP_FUNC, new List<int>());
11             this.greater = new Heap(MIN_HEAP_FUNC, new List<int>());
12             this.median = 0;
13         }
14
15         // O(log(n)) time | O(n) space
16         public void Insert(int number) {
17             if (lowers.length == 0 || number < lowers.peek()) {
18                 this.lower.Insert(number);
19             } else {
20                 this.greater.Insert(number);
21             }
22             this.rebalanceHeaps();
23         }
24     }
25 }
```

Custom Output

Submit Code

```
1 // Do not edit the class below except for
2 // the Insert method. Feel free to add new
3 // properties and methods to the class.
4 public class Program {
5     public class ContinuousMedianHandler {
6         public double median = 0;
7
8         public void Insert(int number) {
9             // Write your code here.
10        }
11
12        public double GetMedian() {
13            return median;
14        }
15    }
16 }
17
```

```
10     return result[0]
11
12     return result[0]
13
14     return result[0]
15
16     return result[0]
17
18     return result[0]
19
20     return result[0]
21
22     return result[0]
23
24     return result[0]
25
26     return result[0]
27
28     return result[0]
29
30     return result[0]
31
32     return result[0]
33
34     return result[0]
35
36     return result[0]
37
38     return result[0]
39
40     return result[0]
41
42     return result[0]
43
44     return result[0]
45
46     return result[0]
47
48     return result[0]
49
50     return result[0]
51
52     return result[0]
53
54     return result[0]
55
56     return result[0]
57
58     return result[0]
59
60     return result[0]
61
62     return result[0]
63
64     return result[0]
65
66     return result[0]
67
68     return result[0]
69
70     return result[0]
71
72     return result[0]
73
74     return result[0]
75
76     return result[0]
77
78     return result[0]
79
80     return result[0]
81
82     return result[0]
83
84     return result[0]
85
86     return result[0]
87
88     return result[0]
89
90     return result[0]
91
92     return result[0]
93
94     return result[0]
95
96     return result[0]
97
98     return result[0]
99
100    return result[0]
```

Run or submit code when you're ready.