Problem 3369: Design an Array Statistics Tracker

Problem Information

Difficulty: Hard
Acceptance Rate: 35.02%
Paid Only: Yes
Tags: Hash Table, Binary Search, Design, Queue, Heap (Priority Queue), Data Stream, Ordered Set

Problem Description

Design a data structure that keeps track of the values in it and answers some queries regarding their mean, median, and mode.

Implement the `StatisticsTracker` class.

* `StatisticsTracker()`: Initialize the `StatisticsTracker` object with an empty array. * `void addNumber(int number)`: Add `number` to the data structure. * `void removeFirstAddedNumber()`: Remove the earliest added number from the data structure. * `int getMean()`: Return the floored **mean** of the numbers in the data structure. * `int getMedian()`: Return the **median** of the numbers in the data structure. * `int getMode()`: Return the **mode** of the numbers in the data structure. If there are multiple modes, return the smallest one.

Note :

* The **mean** of an array is the sum of all the values divided by the number of values in the array. * The **median** of an array is the middle element of the array when it is sorted in non-decreasing order. If there are two choices for a median, the larger of the two values is taken. * The **mode** of an array is the element that appears most often in the array.

Example 1:

Input: ["StatisticsTracker", "addNumber", "addNumber", "addNumber", "addNumber", "getMean", "getMedian", "getMode", "removeFirstAddedNumber", "getMode"] [[], [4], [4], [2],

[3], [], [], [], [], []]

Output: [null, null, null, null, null, 3, 4, 4, null, 2]

Explanation

StatisticsTracker statisticsTracker = new StatisticsTracker(); statisticsTracker.addNumber(4); // The data structure now contains [4] statisticsTracker.addNumber(4); // The data structure now contains [4, 4] statisticsTracker.addNumber(2); // The data structure now contains [4, 4, 2] statisticsTracker.addNumber(3); // The data structure now contains [4, 4, 2, 3] statisticsTracker.getMean(); // return 3 statisticsTracker.getMedian(); // return 4 statisticsTracker.getMode(); // return 4 statisticsTracker.removeFirstAddedNumber(); // The data structure now contains [4, 2, 3] statisticsTracker.getMode(); // return 2

Example 2:

Input: ["StatisticsTracker", "addNumber", "addNumber", "getMean", "removeFirstAddedNumber", "addNumber", "addNumber", "removeFirstAddedNumber", "getMedian", "addNumber", "getMode"] [[], [9], [5], [], [], [5], [6], [], [], [8], []]

Output: [null, null, null, 7, null, null, null, null, 6, null, 5]

Explanation

StatisticsTracker statisticsTracker = new StatisticsTracker(); statisticsTracker.addNumber(9); // The data structure now contains [9] statisticsTracker.addNumber(5); // The data structure now contains [9, 5] statisticsTracker.getMean(); // return 7 statisticsTracker.removeFirstAddedNumber(); // The data structure now contains [5] statisticsTracker.addNumber(5); // The data structure now contains [5, 5] statisticsTracker.addNumber(6); // The data structure now contains [5, 5, 6] statisticsTracker.removeFirstAddedNumber(); // The data structure now contains [5, 6] statisticsTracker.getMedian(); // return 6 statisticsTracker.addNumber(8); // The data structure now contains [5, 6, 8] statisticsTracker.getMode(); // return 5

Constraints:

* `1 <= number <= 109` * At most, `105` calls will be made to `addNumber`, `removeFirstAddedNumber`, `getMean`, `getMedian`, and `getMode` in total. * `removeFirstAddedNumber`, `getMean`, `getMedian`, and `getMode` will be called only if there is at least one element in the data structure.

Code Snippets

C++:

```
class StatisticsTracker {
public:
StatisticsTracker() {

}

void addNumber(int number) {

}

void removeFirstAddedNumber() {

}

int getMean() {

}

int getMedian() {

}

int getMode() {

}
};

/**
* Your StatisticsTracker object will be instantiated and called as such:
* StatisticsTracker* obj = new StatisticsTracker();
* obj->addNumber(number);
* obj->removeFirstAddedNumber();
* int param_3 = obj->getMean();
* int param_4 = obj->getMedian();
* int param_5 = obj->getMode();
*/
```

Java:

```
class StatisticsTracker {

public StatisticsTracker() {

}

public void addNumber(int number) {

}

public void removeFirstAddedNumber() {

}

public int getMean() {

}

public int getMedian() {

}

public int getMode() {

}
}

/**
* Your StatisticsTracker object will be instantiated and called as such:
* StatisticsTracker obj = new StatisticsTracker();
* obj.addNumber(number);
* obj.removeFirstAddedNumber();
* int param_3 = obj.getMean();
* int param_4 = obj.getMedian();
* int param_5 = obj.getMode();
*/
```

Python3:

```
class StatisticsTracker:
```

```
def __init__(self):


def addNumber(self, number: int) -> None:


def removeFirstAddedNumber(self) -> None:


def getMean(self) -> int:


def getMedian(self) -> int:


def getMode(self) -> int:



# Your StatisticsTracker object will be instantiated and called as such:
# obj = StatisticsTracker()
# obj.addNumber(number)
# obj.removeFirstAddedNumber()
# param_3 = obj.getMean()
# param_4 = obj.getMedian()
# param_5 = obj.getMode()
```