Problem 1146: Snapshot Array

Problem Information

Difficulty: Medium
Acceptance Rate: 36.67%
Paid Only: No
Tags: Array, Hash Table, Binary Search, Design

Problem Description

Implement a SnapshotArray that supports the following interface:

* `SnapshotArray(int length)` initializes an array-like data structure with the given length. **Initially, each element equals 0**. * `void set(index, val)` sets the element at the given `index` to be equal to `val`. * `int snap()` takes a snapshot of the array and returns the `snap_id`: the total number of times we called `snap()` minus `1`. * `int get(index, snap_id)` returns the value at the given `index`, at the time we took the snapshot with the given `snap_id`

Example 1:

Input: ["SnapshotArray","set","snap","set","get"] [[3],[0,5],[],[0,6],[0,0]] **Output:** [null,null,0,null,5] **Explanation:** SnapshotArray snapshotArr = new SnapshotArray(3); // set the length to be 3 snapshotArr.set(0,5); // Set array[0] = 5 snapshotArr.snap(); // Take a snapshot, return snap_id = 0 snapshotArr.set(0,6); snapshotArr.get(0,0); // Get the value of array[0] with snap_id = 0, return 5

Constraints:

* `1 <= length <= 5 * 104` * `0 <= index < length` * `0 <= val <= 109` * `0 <= snap_id < `(the total number of times we call `snap()`) * At most `5 * 104` calls will be made to `set`, `snap`, and `get`.

Code Snippets

C++:

```
class SnapshotArray {
public:
SnapshotArray(int length) {

}

void set(int index, int val) {

}

int snap() {

}

int get(int index, int snap_id) {

}
};

/**
* Your SnapshotArray object will be instantiated and called as such:
* SnapshotArray* obj = new SnapshotArray(length);
* obj->set(index,val);
* int param_2 = obj->snap();
* int param_3 = obj->get(index,snap_id);
*/
```

Java:

```
class SnapshotArray {

public SnapshotArray(int length) {

}

public void set(int index, int val) {

}

public int snap() {
}

public int get(int index, int snap_id) {

}
}

/**
* Your SnapshotArray object will be instantiated and called as such:
* SnapshotArray obj = new SnapshotArray(length);
* obj.set(index,val);
* int param_2 = obj.snap();
* int param_3 = obj.get(index,snap_id);
*/
```

Python3:

```
class SnapshotArray:

def __init__(self, length: int):


def set(self, index: int, val: int) -> None:


def snap(self) -> int:


def get(self, index: int, snap_id: int) -> int:



# Your SnapshotArray object will be instantiated and called as such:
# obj = SnapshotArray(length)
# obj.set(index,val)
# param_2 = obj.snap()
# param_3 = obj.get(index,snap_id)
```