Question: https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed/submissions/

For this question, we can use an arraylist keep track of numbers, and a hashMap in which key is the number and value is a hashset which contains all the indices of the given value in the array.

Whenever we need to remove a value, we should swap the indices of the last integer and given value in the arraylist, so that we can on1y remove the tail and the time complexity is O(1).

Whenever we need to add a value, add the value to the end of the array (P.s. O(1), and push the number and the corresponding position (the end of the array) to the map. (O(1)).

We could use the random() function when we need to retrieve an element from the array haphazardly.

Code:  
class RandomizedCollection {

HashMap<Integer, HashSet<Integer>> map = new HashMap<>();

ArrayList<Integer> elements = new ArrayList<>();

/\*\* Initialize your data structure here. \*/

public RandomizedCollection() {

map = new HashMap<>();

elements = new ArrayList<>();

}

/\*\* Inserts a value to the collection. Returns true if the collection did not already contain the specified element. \*/

public boolean insert(int val) {

elements.add(val);

if(map.containsKey(val)){

HashSet<Integer> poss = map.get(val);

poss.add(elements.size() - 1);

return false;

}

else{

HashSet<Integer> poss = new HashSet();

poss.add(elements.size() - 1);

map.put(val, poss);

return true;

}

}

/\*\* Removes a value from the collection. Returns true if the collection contained the specified element. \*/

public boolean remove(int val) {

if(map.containsKey(val)){

HashSet<Integer> poss = map.get(val);

int index = poss.iterator().next() ;

int finalEle = elements.get(elements.size() - 1);

if(val != finalEle){

HashSet<Integer> finalValuePoss = map.get(finalEle);

finalValuePoss.add(index);

finalValuePoss.remove(elements.size() - 1);

elements.set(index, finalEle);

poss.remove(index);

elements.remove(elements.size() - 1);

}

else{

poss.remove(elements.size() - 1);

elements.remove(elements.size() - 1);

}

if(poss.size() == 0){

map.remove(val);

}

return true;

}

else

return false;

}

/\*\* Get a random element from the collection. \*/

public int getRandom() {

return elements.get((int)(Math.random()\*elements.size()));

}

}

/\*\*

\* Your RandomizedCollection object will be instantiated and called as such:

\* RandomizedCollection obj = new RandomizedCollection();

\* boolean param\_1 = obj.insert(val);

\* boolean param\_2 = obj.remove(val);

\* int param\_3 = obj.getRandom();

\*/

Github Link :<https://lnkd.in/ecwtJeaz>