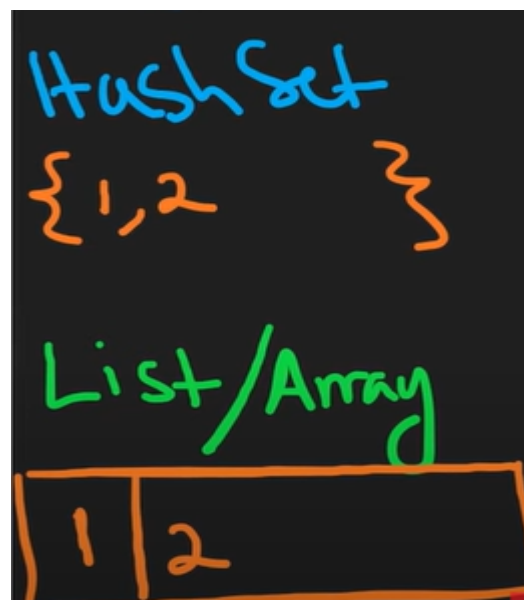
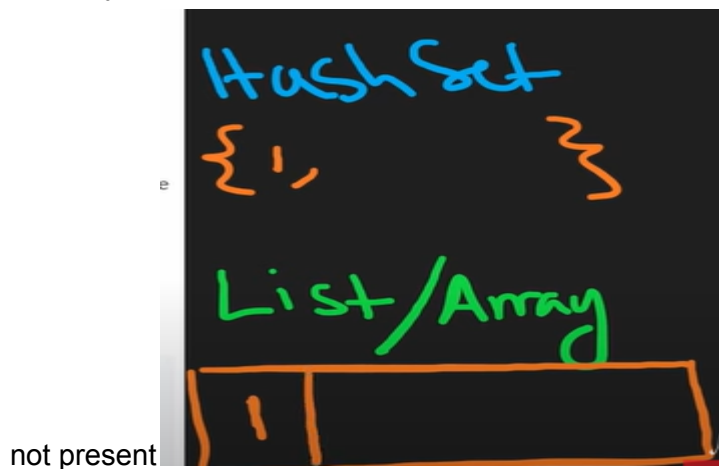


## 380. Insert Delete GetRandom O(1)

```
RandomizedSet() Initializes the RandomizedSet object.  
• bool insert(int val) Inserts an item val into the set if not present. Returns true if the item was not present, false otherwise.  
• bool remove(int val) Removes an item val from the set if present. Returns true if the item was present, false otherwise.
```

1. Here for inserting and removal..we'll be using hashset...which gives  $O(1)$
2. But the problem arises..while getting a random value ..because hashset cannot have index..and we cannot get the random value...
3. So here we use both hashset and a list
4. Example
5. Lets say we want to insert 1..then we call insert() function ..and we return true as it was



6. Now we insert 2 we call insert() function and it returns true
7. Now if we want to find random value....we can get easily from the list

8. Now if we want to remove 2..then we can easily remove it..as it was in the end of the list
9. But we cannot remove a middle element in  $O(1)$ ..so we use hashmap here instead of hashset
10. If the val is present in the hashmap...and we retrieve the index of the val using hashmap
11. Next we'll retrieve the last element of the list
12. Now we'll swap the last element of the list with **val** which we are deleting
13. Now val will be at last of list..and we use pop() to remove it
14. We'll update the index of lastVal(swapped) = idx
15. Atlast we'll del the val in numMap

Code:

```
def __init__(self):
    self.numMap = {}
    self.numList = []

def insert(self, val: int) -> bool:
    res = val not in self.numMap
    if res:
        self.numMap[val] = len(self.numList)
        self.numList.append(val)
    return res

def remove(self, val: int) -> bool:
    res = val in self.numMap
    if res:
        idx = self.numMap[val]
        lastVal = self.numList[-1]
        self.numList[idx] = lastVal
        self.numList.pop()
        self.numMap[lastVal] = idx
        del self.numMap[val]
    return res

def getRandom(self) -> int:
    return random.choice(self.numList)
```

1.