Today's Content

- → HashMap Intro
- → frequency of each query
- → first non repeating element
- → distinct elements
- → subarray with sum = 0.
- → Strings fact.

Quote :-

Choose the non-emotional response to any given situation and see how much easier your life becomes.

→ Doubt Session

String str = "harish"

str = str + 'a' O(n).

String are immutable

↓

Java, Python.

↓

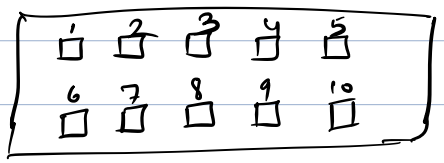
StringBuilder. StringBuffer.

```
arr[n]
str = "";
for (i=0; i<n; i++) {
    str += arr[i];
}
```

O(n) → O(n²).

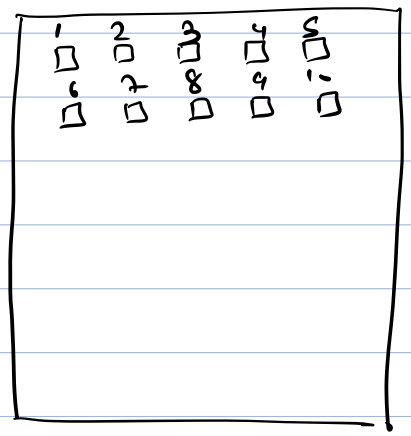
HashMap Intro :-

a) Shriram + Rakesh → Registery.



Room No	Available
1	✗
2	✗
3	✓
4	✗

b)



→ Room Number [1 1000]

bool room[1001]

room[1] = false

room[50] = false.

c) Covid Came → Now in 10yrs.

1000 lucky numbers.

range [1 to 10^9]

arr[10^9]

↳ waste of space.

HashMap <key, Value>

<79, occupied>

<87, not occupied>

<1001, occupied>

T.C → O(1)

S.C → O(N).

key :- should be unique

value :- can be anything.

Q1) Store population of every country.

key : country name -> String

value : Population -> Int

HashMap<String, Int> hm

Q2) No. of States in each country.

key -> country name -> String

values -> No. of States -> Int

HashMap<String, Int> hm

Q3) for every country we want to know all state names.

key -> country name -> String

Value -> name of States -> List<String>

HashMap<String, List<String>>

-> Dynamic Array.

↳ C++ -> vector

↳ Python -> List

↳ Java -> ArrayList

Q4) for every country store population of each State

key -> countryName -> String

Value -> population of each state -> HashMap<String, Int>

HashMap<String, HashMap<String, Int>>

key value.

Obs1:- Value can be anything.

Obs2:- key can be only primitive data type.

int / long / float / double / String / char

key	key - value
India	Andhra - 500 Telangana - 1000 Karnataka - 1600
USA	Texas - 60 Cal - 50 New - 70

HashMap functionality

<key, value>

size() → no of keys
insert (key, value)
search (key)
delete (key)
update (key, value)

HashMap
<India, 800>
<USA, 790>
<India, 810>

overridden.

when we insert, same key again it will override previous val

HashSet

<keys>

size()
insert (key)
search (key)
delete (key)

Note :- A single operation in hashmap/hashset => O(1)

→ Order is not ensured in hashmap/hashset.

→ C# hashing library name in diff languages

Pseudocode	Java	C++	Python	JS
HashMap	HashMap	unordered_map	dictionary	map
HashSet	HashSet	unordered_set	set	Set

C#
Dictionary
HashSet

10) find frequency of numbers.

Given N array elements & Q queries, for each query we have to find freq. of that ele. in array.

Constraints

$1 <= N <= 10^5$, $1 <= Q <= 10^5$

arr[10] = { 2, 6, 3, 8, 2, 8, 2, 3, 8, 10, 6 }

10 → 1
8 → ~~1~~ ~~2~~ 3
2 → ~~1~~ ~~2~~ 3
6 → ~~1~~ 2
3 → ~~1~~ 2

Q: 4

2 : 3
8 : 3
3 : 2
5 : 0

idea 1:- for every query iterate & get count

T.C → Q × N, S.C → O(1).

idea 2:- Use a hashmap

2 → 3
6 → 2
3 → 2
8 → 3
10 → 1

key → array element → int
value → freq. of element → int
HashMap < int, int > hm;

```
void PrintFreq(int arr[], int Q[]) {

    int n = arr.length;
    int m = Q.length;
    HashMap<int, int> hm;
    for (i=0; i<n; i++) {                    // O(n)
        if (hm.search(arr[i]) == True) {
            hm[arr[i]] += 1;
        } else {
            hm[arr[i]] = 1
        }
    }

    for (i=0; i<m; i++) {                    // O(m)
        if (hm.search(Q[i]) == True) {
            print(hm[Q[i]])
        } else {
            print(0)
        }
    }
}
```

T.C → O(n+q)

S.C → O(n)

Break 10:16 - 10:26 pm

Q) find the first non-repeating element,

↳ first element from start, non-repeating.

arr[6] = {1, 2, 3, 1, 2, 5}, ans = 3

arr[7] = {2, 6, 8, 4, 7, 2, 9}, ans = 6

arr[8] = {4, 3, 3, 2, 5, 6, 4, 5}, ans = 2.

Idea 1:- ~~1)~~ Insert all elements in hmap & iterate on hashmap to get 1st key with val = 1.

arr[6] = {1, 2, 3, 1, 2, 5}

Hashmap

<1,2>, <5,1>, <2,2>, <3,1>

Note:- Order of insertion of keys is not maintained.

2) Insert all elements in hmap. &, Iterate on array & get first element with freq = 1.

Step1:- Insert all elements in map. T.C→O(n)
Step2:- Iterate on arr[] and get T.C→O(N) 1st element with freq 1.

T.C → O(n)
S.C → O(n)

Todo. ↑ Code

Ques) Given arr[N] elements, find no. of distinct elements.

arr[5] = {3, 5, 6, 5, 4}, ans = 4
arr[5] = {1, 1, 1, 2, 2}, ans = 2
arr[3] = {3, 3, 3}, ans = 1.

idea :- insert all elements in hset.

arr[7] = {6, 3, 7, 3, 8, 6, 9}
Hashset<int> hs;

{6, 3, 7, 8, 9} → hs.size() → 5.

Note :- In hashset, if same key is inserted, multiple times, we will still have 1 occurence

Pseudocode :-

```
Hashset<int> hs;                    T.C -> O(N)
for (i=0; i<n; i++) {               S.C -> O(N)
    hs.insert(arr[i]);
}
return hs.size();
```

Ques) Given arr[N] elements, check if all elements are distinct or not.

arr[5] = { 6, 8, 3, 2, 73 } = return True

arr[7] = { 3, 1, 6, 1, 4, 9, 63 } = return false.

idea :- insert all elements in hset;

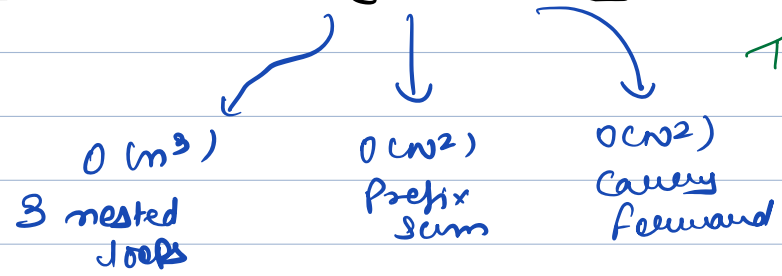
```
if (hashset.size == N) {
    // all elements are distinct
    return True;
} else {
    return false.
}
```

T.C -> O(N)
S.C -> O(N).

Q) Given arr[N] elements, check if there exists a subarray with sum = 0.

	0	1	2	3	4	5	6	7	8	9
arr[10] =	2	2	1	-3	4	3	1	-2	-3	2

idea :- for every subarray calculate sum == 0.



	0	1	2	3	4	5	6	7	8	9
arr[10] =	2	2	1	-3	4	3	1	-2	-3	2
pf[10] =	2	4	5	2	6	9	10	8	5	7

Obs :- 1) In pfSum array if value repeats it means there is a subarray with sum = 0.

2) If you have a 0 in prefix array the also, subarray sum = 0.

arr[] = {2, -5, 3, 6}

pf[] = {2, -3, 0, 6}

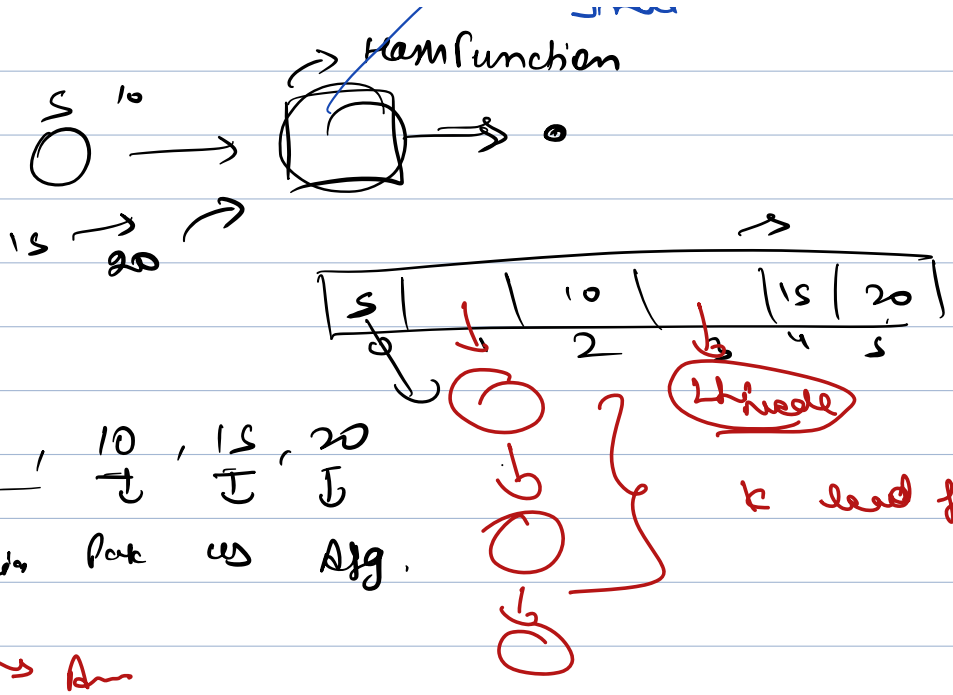
→ T.C → O(N)

```
bool subarrayZero(int arr[]) {
    int n = arr.length;          S.C → O(N)
    int pf[n]; // Todo constructing it.

    HashSet<int> hs;
    for (i=0; i<n; i++) {
        if (pf[i]==0) {
            return True
        }
        hs.insert(pf[i])
    }

    if (hs.size()<N) { //repetition in Pf[]
        return True;
    } else {
        return false;
    }
}
```

Gradia
Pakistan
Sindi



2015.

ArrayList<String> ll = new ArrayList<>();

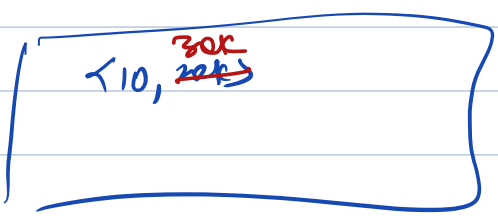
ll.add("India");

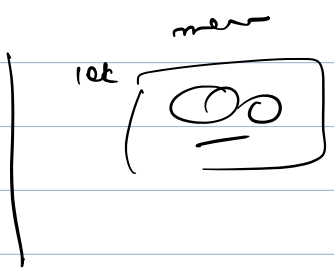
list.add("USA");

2015 ll2 = ("Kunal", "Sriram");

HashMap<Int, List<String>>

map.put(10, ll); → map.put(10, ll2);





ll = 101

```
ArrayList<String> ll = new ArrayList<>();
ll.add("India");
ll.add("USA");

ll = new ArrayList<>();
```