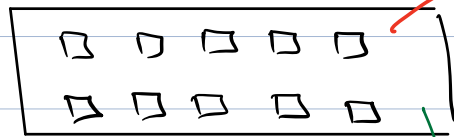


Hashing Intro :-

Abhir up - Diya.



1000 rooms

Dhyanashree ji

Program

covid
(status)

bool ch[1001] = {T, F}

indicates
availability

// books room \rightarrow 5

ch[5] = f.

$x \rightarrow$ ch[x] = True \rightarrow avail

ch[x] = False \rightarrow unavail.

Tina's

[3, 5, 7, 9 ... 99, 129 ...]

\rightarrow [1 to 10^8]

Sol:-

bool ch[10^8+1] = {T, F}

2 issues:-

\rightarrow Memory wastage

\rightarrow 10^8 byte \approx 100 MB.

Dictionary.
HashMap :- $\langle \text{key}, \text{value} \rangle$

↓
keys have to
be distinct.
 $\langle 30, \text{occupied} \rangle$
 $\langle 198, \text{occupied} \rangle$
 $\langle 10^6, \text{occupied} \rangle$

Advantage:-

Data Retrieval is very fast.

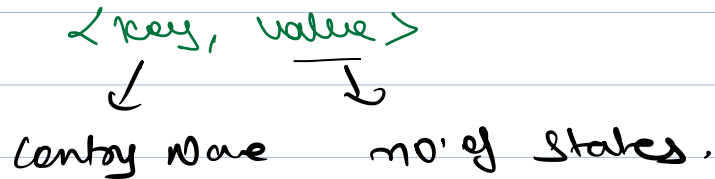
HashMap → Inbuilt library,
↳ Data Retrieval / Searching /
Insertion / updates.

Q! Store population of every country

$\langle \text{key}, \text{value} \rangle$
↓ ↓
Country name Population

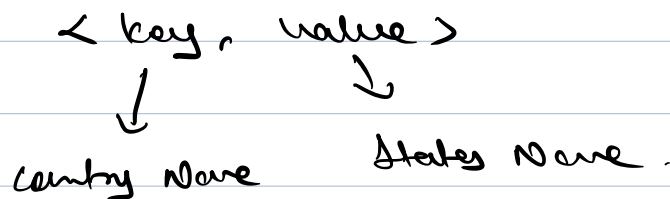
→ HashMap $\langle \text{String}, \text{long} \rangle$

Q2 No: of States in each country



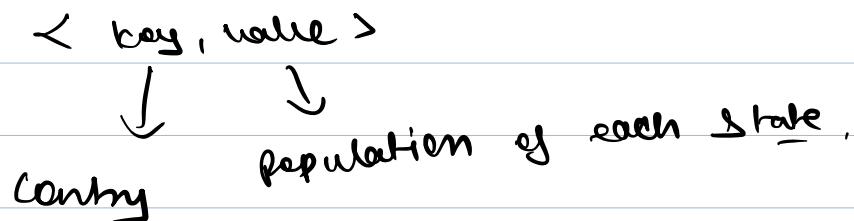
HashMap < String, Int >

Q3 For every country we want to store all state name



HashMap < String, List >

Q4 For every country store population of each state.



HashMap < String, HashMap < String, Int > >

obs:- 1) value can be anything.

{ int, long, string, float, double }

Can key type be an array list? - No

HashMap <key, value>

insert <key, value>

Search <key>

Delete <key, value>

update value of a key.

update a key - ? X

↳ not possible.

size ();

Name and phone no. in map.

<Ravi, 1234>

<Mangesh, 123>

<Vishal, 2457>

<Diya, 9810>

After inserting, if we print all data,
order is different >

HashSet <keys>

Ans $\rightarrow O(1)$.

Ques 1) Given an array and Q queries:-

arr[] \rightarrow { 2, 6, 3, 8, 2, 8, 2, 3, 8 }

for each query element tell the freq.

0 : 3

2 : 3

8 : 3

3 : 2

App 1) For every query iterate & check,

for ($i = 0; i < Q; i++$) { \rightarrow 0

// Read the query, n .

Count the freq of n \rightarrow N

3

$O(Q * N)$

App 2:- Store freq in map.

\rightarrow $\text{map}(\text{freq})$

<key, value>

\downarrow

$\text{map}(\text{array elements})$

arr[] → { 2, 6, 3, 8, 2, 8, 2, 3, 8, 3 }

<2,3>
<6,1>
<3,2>
<8,3>

Pseudocode :-

```
HashMap<int, int> hm;  
int freq = hm.get(arr[i])  
hm.update(arr[i], freq+1) if (arr[i] is in hm) {
```

hm[arr[i]]++

<2,3>
<6,1>
<3,2>
<8,3>

}

else {

hm.insert(arr[i], 1)

}

hm.put(arr[i], 1)

arr[] → { 2, 6, 3, 8, 2, 8, 2, 3, 8, 3 }

```
for (i = 0; i < n; i++) {
    // x in query.
```

T.C $\rightarrow O(N+Q)$.
S.C $\rightarrow O(N)$.

```
print(hm[x]);
}
```

// java

HashMap<key, value> hm =

new HashMap<>();

Ques) find the first non-repeating element.

Ex 1 :- arr[6] :- {1, 2, 3, 1, 2, 5}
arr[7] :- {4, 8, 8, 3, 4, 2, 3}



5	→	1
1	→	2
3	→	1
2	→	2



Step 1:- Build HM.

Step 2:- Iterate the array and
get 1st element whose
freq = 1.

```
for (i = 0; i < n; i++) {
    freq = hm[arr[i]]
    if (freq == 1) {
        return arr[i];
    }
}
```

10:18 - 10:28 pm.

Ques 1) Given N array elements, find no. of distinct elements.

arr[7] = { 6, 3, 7, ^{*}3, 8, ^{*}6, 9 } \rightarrow 5.

arr[8] = { 2, 9, 8, 7, 6, 8, 2, 7, 3 } → 5.

 $\langle 2, 2 \rangle$ $\langle 9, 1 \rangle$

$\angle 812$

 $\langle 7, 2 \rangle$ $\langle 6, 1 \rangle$

→ hashmap size.

T.C $\rightarrow O(n^2)$

2

9

8

7

6

$T.C \rightarrow O(n^2)$

→ test - niz!


```

HashSet<Integer> hs;
for (i=0; i<n; i++) {
    hs.insert(arr[i]);
}

print(hs.size());

```

(If we store same keys again in HashSet, it will only store a single occurrence).

kgf-1

kgf-2

Ques :- Given N array elements, check if there exists a subarray with sum = 0.

	0	1	2	3	4	5	6	7	8	9
arr :-	2	2	1	-3	4	3	1	-2	-3	2
PF :-	2	4	5	2	6	9	10	8	5	7

5

5

These elements sum to 0.

$$PF[2] = sum[0-2] = 5$$

$$5$$

$$PF[8] = sum[0-8] = 5$$

$$PF[8] = sum[0-2] + sum[3-8] \Rightarrow PF[8] = PF[2] + sum[3-8]$$

Brute force:-

1) Generate all subarray

sums

↓
 $O(N^3)$

→ carry forward

↓
 $O(N^2)$

$$\text{sum}[3-8] = \text{pf}[8] - \text{pf}[2]$$

$$\text{sum}[3-8] = \underline{0}$$

Obs:- ① if your pf(), if an element repeats, subarray sum = 0.

② check if 0 is in pf[].

	0	1	2	3
En:-	3	-1	-2	4
pf:-	3	2	0	4

Pseudocode:-

Steps:-

- 1) calculate pf[].
- 2) check if 0 is present in pf[].
- 3) check if element repeats in pf[].

↳ do it with prev ques.



In the last ques end check:-

(HashSet.size() == n) or not.

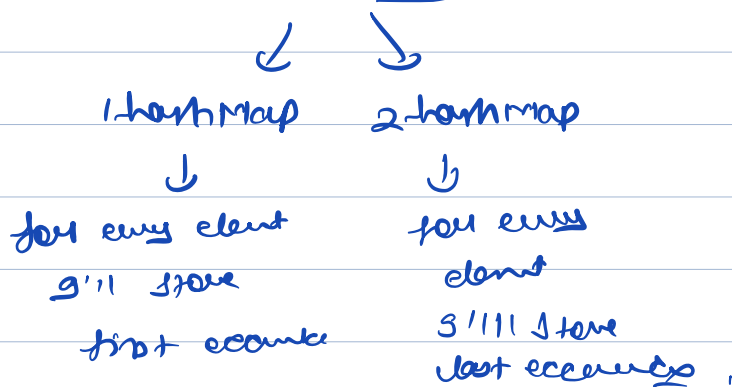
Ques) Find the length of longest subarray with sum = 0.

	0	1	2	3	4	5	6	7	8	9	10	11	12
arr[] =	3	3	4	-5	-2	2	1	-3	3	-1	5	-4	-1
pf[] =	3	6	10	5	3	5	6	3	6	5	10	6	5

3 6 10 5 3 5 6 3 6 5 10 6 5

Obs:- for every element in pf(), store its 1st and last occurrence.

2 hashmaps:-



pf [] = 3 6 10 5 3 5 6 3 6 5 10 6 5

	first index	len	ans
<3, 0>	0	4	4
<6, 1>	3	2	4
<10, 2>	1	6	5
<5, 3>	0	7	7
	1	7	7
	3	6	7
	2	8	8
	1	10	10
	3	9	<u>10</u>

hm <int, int> ,

hm.insert <0, -1> → Edge case.

for (i = 0; i < n; i++) {

```

    if (pf [i] is in hm) {
        int len = i - hm.get (pf [i])
        ans = max (ans, len);
    }
    else {
        hm.insert (pf [i], i);
    }
}

```

3 3

Edge Case :-

	0	1	2	3	4	5
arr \rightarrow	4	-3	-1	2	3	-5
pf \rightarrow	4	1	0	2	5	0

$\langle 0, -1 \rangle$

Ques) Given array only contains 0's and 1's,
find max length subarray
which contains equal 0's and 1's.

Soln:- replace all 0 \rightarrow with -1.

\rightarrow 0 0 1 1 0 0 1 1 1

\downarrow

-1 -1 1 1 -1 -1 1 1 1

double :-

check if `arr[i]` is in `hmap` or not.

if (`hmap.containsKey(arr[i])` == `True`) {

}

a b c d e f ,