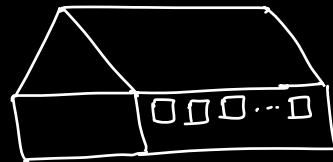


- Why HashMap? ✓
- How HashMap is implemented?
- Problem patterns. ✓

L.L  
BST's  
B.B.S.T.

Akshay & Tanya  
2018: No HashMap

Register.



1000 rooms

R.no	Status
1	0
2	0
3	0
...	...
101	0
102	0
...	...
1000	0

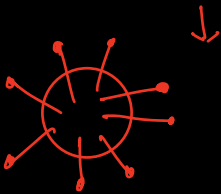
⇒ int arr[1001]

0	1	2	3	4	...	1000
x	0					

0 ⇒ free

1 ⇒ occupied.

TC: update/ the status  
get L arr[i] = 1/0 ⇒ O(1)



Covid

Numerologist

random 1000 no's  
in the range of  $[1, 10^9]$

[42, 58, 110, 189, ... 1009, ...  $10^9$ ]

↓  
index

int arr[ $10^9 + 1$ ]

0  $\Rightarrow$  free

1  $\Rightarrow$  occupied.

Memory wastage :-

Size of the Array =  $10^9$

No. of rooms = 1000

$$\begin{aligned} \text{memory wastage} &= 10^9 - 10^3 \\ &= \begin{array}{r} 1000000000 \\ 1000 \\ \hline \end{array} \end{aligned}$$

$\approx 10^9$

Size of int = 4 B

$$\begin{aligned} \text{Total memory} &= 10^9 \times 4 \text{ B} \\ &= \underline{\underline{4 \text{ GB}}} \end{aligned}$$

NOT A Good approach.

Requirements

1) No memory wastage.

2) Check()

3) update()

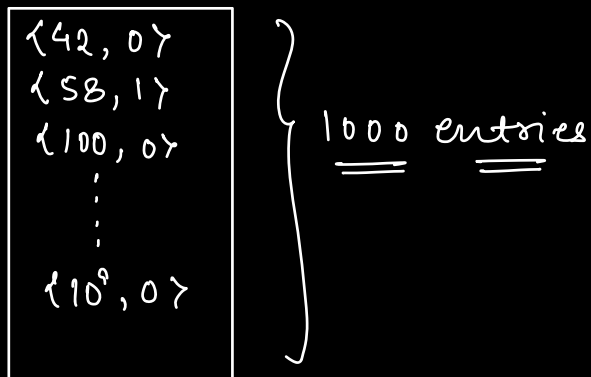
$O(1)$  avg case

Operations

1 : 2s  
2 : 2s  
3 : 4s  
4 : 1s  
5 : 1s

$$\left. \begin{array}{l} 1 : 2s \\ 2 : 2s \\ 3 : 4s \\ 4 : 1s \\ 5 : 1s \end{array} \right\} \text{avg TC : } \frac{10}{5} = \underline{\underline{2 \text{ sec}}}$$

HashMap  $\{K, V\}$



HashMap { Key, Value }

Astrologist  
=  $\hookrightarrow$  Room Name (String)

$\Rightarrow$  Array sol<sup>n</sup> won't work.

$\Rightarrow$  HashMap  $\{ \textcircled{K}, V \}$

Keys should be unique in the HM

Room Name (String)      Status (int | bool)

Java  
HashMap  
HashSet

Python/C#  
Dict  
set

C++  
unordered\_map  
unordered\_set

JS/Python  
map  
Set

Quiz

Country → Population  
↓ ↓  
String long

HashMap (String, long)

Quiz

Country → No. of states  
↓ ↓  
String int

HashMap (String, int)

↓ ↓  
Country no. of states.  
name

Quiz

Country → Name of all states.

India → "Haryana", "Punjab", "Delhi", ...  
(key) (value)

HashMap (String, List (String))

Quiz

Country → Population of all states.

India → { Haryana : 100  
Maharashtra : 120  
Punjab : 110  
⋮ }

HashMap (String, HashMap (String, long) )

## HashMap Functions

- 1) Check(k) / contains(k)
- 2) Update(k, v)
- 3) insert(k, v)
- 4) delete(k)
- 5) size()  
↳ no. of keys.
- 6) get(k) → value.

TC:  $O(1)$  avg case.

Q. Given an Array of size N & Q queries. for every query → int n.  
return the frequency of n for each query.

A: [ 2, 6, 3, 8, 2, 8, 2, 3, 8 ]

Q=4

- 1)  $n = 2 \Rightarrow 3 \Rightarrow O(N)$
- 2)  $3 \Rightarrow 2$
- 3)  $4 \Rightarrow 0$
- 4)  $8 \Rightarrow 3$

Brute force

TC:  $O(Q \cdot N)$

SC:  $O(1)$

⇒ HashMap { int, int }  
          ↓      ↓  
          A[i]  freq.

A: [ 2, 6, 3, 8, 2, 8, 2, 3, 8, 10000 ]

{ 2, 3 }  
{ 6, 1 }  
{ 3, 2 }  
{ 8, 3 }

map

```
for( i = 0; i < N; i++ ) {  
    if( map.containsKey(A[i]) ) {  
        map.update(A[i], map.get(A[i]) + 1);  
    }  
    else {  
        map.insert(A[i], 1);  
    }  
}
```

Annotations:  $O(1)$  for containsKey,  $O(1)$  for update,  $O(1)$  for insert.

TC:  $O(N)$

SC:  $O(N) \rightarrow$  HashMap

\* for every query: n  
    return map.get(n) }  $O(Q)$   
                            ↓  $O(1)$

Total TC:  $O(N) + O(Q) \Rightarrow O(N+Q)$

SC:  $O(N)$

Q. Given an Array, Count the no. of distinct elements in the Array.

A: [7, 3, 2, 1, 3, 7, 0]  $\Rightarrow$  5

↑ ↑ ↑ ↑ ↑ ↑ ↑  
✓ ✓ ✓ ✓ × × ✓

Quiz

A: [6, 3, 7, 3, 8, 6, 4]

$\Rightarrow$  5

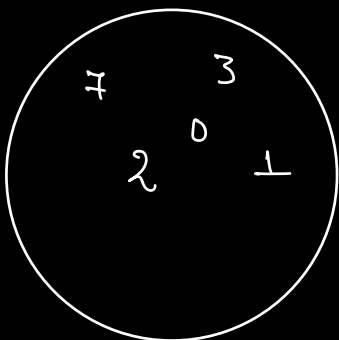
A: [7, 3, 2, 1, 3, 7, 0]

{7, 2}  
{2, 1}  
{0, 1}  
{1, 1}  
{3, 2}

$\Rightarrow$  return map.size();

HashSet | Set  $\Rightarrow$  Unique keys.

A: [7, 3, 2, 1, 3, 7, 0]  
↑



$\Rightarrow$  return set.size();

insert(k)  
delete(k)  
Contains(k)  
Size(k)

} O(1) avg case

TC:  $O(N)$   
SC:  $O(N)$

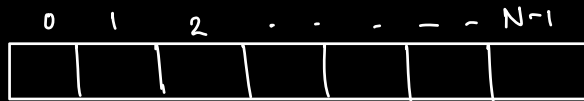
Q. Given an Array of size  $N$ , check if there exists a subarray with  $\text{sum} = 0$

Amazon  
Adobe  
MSI  
linkedIn  
...

A: [2, 2, 1, -3, 4, 3, 1, -2, -3]

$\Rightarrow$  True.

No. of subarrays in an Array of size =  $N$ .



$N + (N-1) + (N-2) + \dots + 1$

$$\frac{N(N+1)}{2} \Rightarrow \underline{O(N^2)}$$

```
for (s = 0; s < N; s++) {  
    for (e = s; e < N; e++) {  
        sum = 0  
        for (i = s; i <= e; i++) {  
            sum += A[i]  
        }  
        if (sum == 0)  
            return true;  
    }  
}  
return false;
```



TC:  $O(N^3)$

SC:  $O(1)$

# Using PS

```
for (s = 0; s < N; s++) {  
    for (e = s; e < N; e++) {  
        sum = ps[e] - ps[s-1];  
        if (sum == 0)  
            return true;  
    }  
}  
return false;
```

TC:  $O(N^2)$

SC:  $O(N)$

↳ PS Array

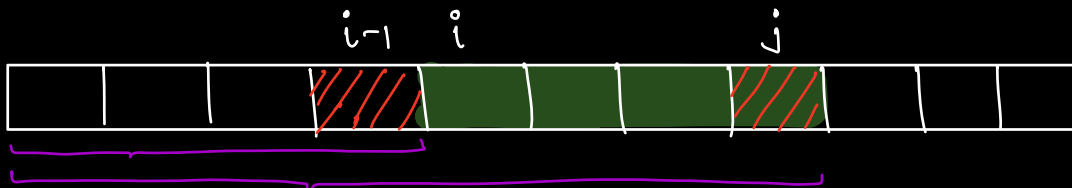
#

$sum(i, j) = ps[j] - ps[i-1]$

if  $sum(i, j) = 0$

$ps[j] - ps[i-1] = 0$

$\Rightarrow ps[j] = ps[i-1]$



$$PS[i-1] = \text{sum}(0, i-1)$$

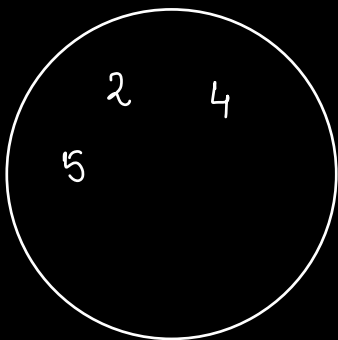
$$PS[j] = \text{sum}(0, i-1) + \text{sum}(i, j)$$

$$\cancel{\text{sum}(0, i-1)} = \cancel{\text{sum}(0, i-1)} + \text{sum}(i, j)$$

$$\boxed{\text{sum}(i, j) = 0}$$

⇒ If there are duplicate values in the PS array then there will be a subarray with sum=0

A: [ 2, 2, 1, -3, 4, 3, 1, -2, -3 ]  
 PS: 2 4 5 2 6 9 10 8 5  
 ↑



```
for( i = 0; i < N; i++ ) {
    if( set.contains(a[i]) )
        return true;
    else {
        set.insert(a[i]);
    }
}
```

A: [1, 2, 0, 8]  
PS: [1, 3, 3, 11]

✓

A: [0, 3, 2, 8]  
PS: [0, 3, 5, 13]

if (A[i] == 0)  
return true;

A: [3 -1 -2 4]  
PS: [3 2 0 4]  
↑

if (PS[i] == 0)  
return true;

TC:  $O(N)$

SC:  $O(N) + O(N) \Rightarrow \underline{\underline{O(N)}}$   
PS Set

Q Does HashMap / HashSet preserve order of elements.

⇒ NO

— \* —