

Hashing - 1

"EKla chalo re"
by
Amjad Ali Khan
Sahab

विद्यां ददाति विनयं, विनयाद् याति पात्रताम् ।
पात्रत्वात् धनमाप्नोति, धनात् धर्मं ततः सुखम् ॥

vidyā dadāti vinayam, vinayād yāti pātratām |
pātratvāt dhanamāpnoti dhanāt dharmam tataḥ sukham ||

"Knowledge gives us discipline,
from discipline comes competence,
from competence comes wealth,
from wealth comes right actions,
and from right actions, one gets happiness."

①

Revision class:

→ for all

Reattempt contest-2!

48 hours.

tomorrow, Saturday [3 PM - 5 PM]

optional

recording will be available.

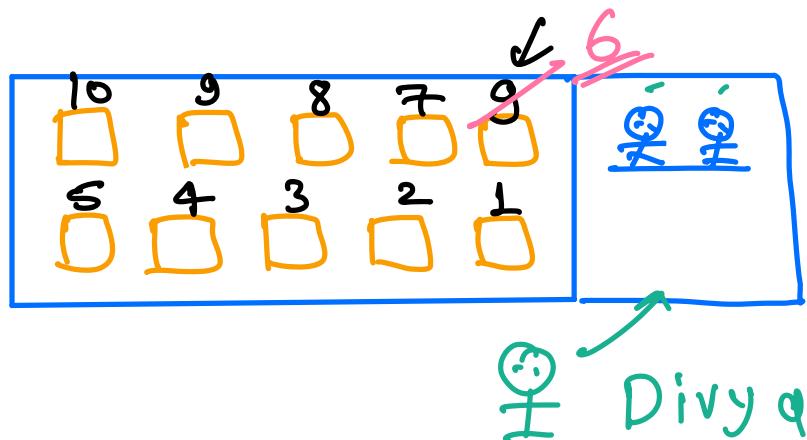
Saturday 5 PM - Monday 5 PM

→ Before attempting quiz, please watch
the contest discussion class recording

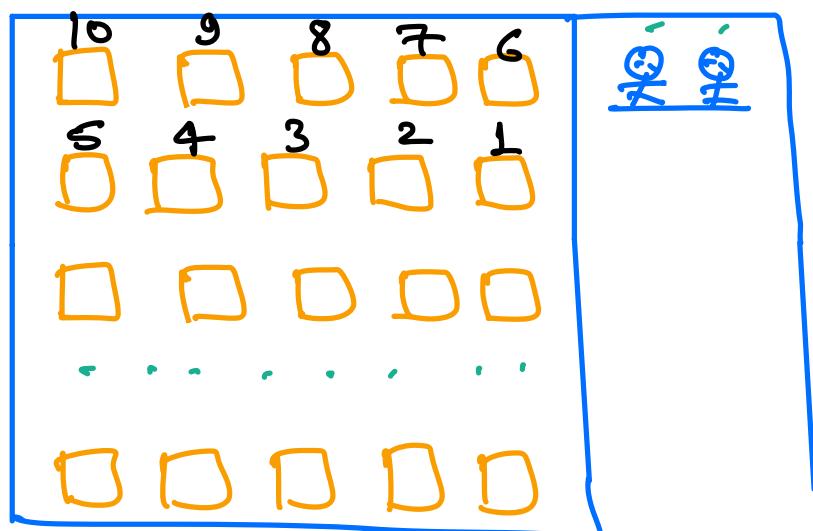
Contest 1 & contest-2 from Sat 5 PM - Mon 5 PM

↓
next weekend.

~~(*)~~ Faisal, vitul. Hotel businen.



Room No.	Available
1	✓
2	✗
:	
9	✓
10	✓



Room No.	Available
1	✓
2	✗
3 9 2 5	
S999 6000	
⋮	
	✗

⇒ which data structure we can use to digitise this information:

isRoomAvailable \Rightarrow 
 $\underline{6001}$ $\text{isRoomAvailable}[\text{RoomNo}] = \text{true/false}$ + fast $O(1)$

⇒ lucky numbers for your rooms::

$[1 \dots 1000]$

$[1, 9, 7, 97, 118, \dots 1001] .$

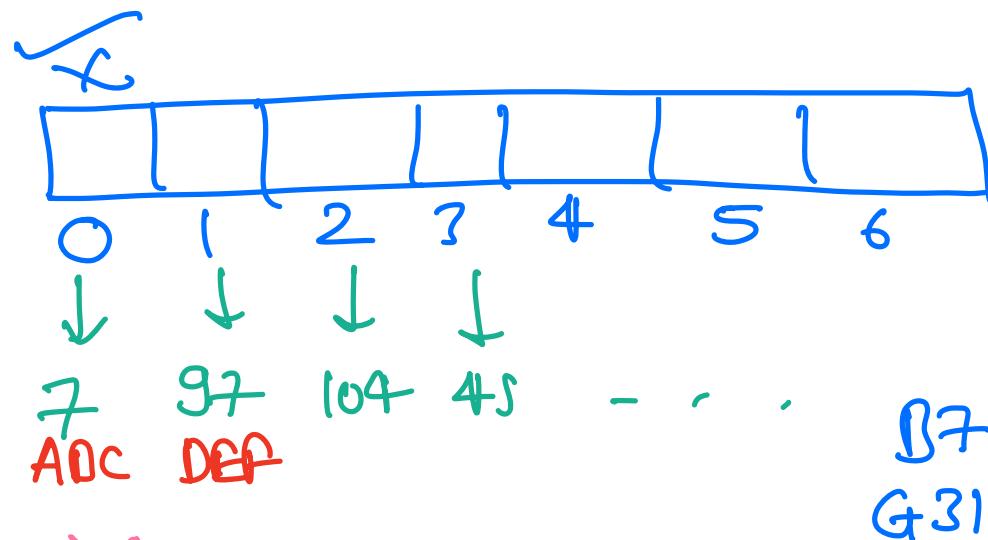
$\overbrace{1000}^{\text{1000}} \Rightarrow [\underline{1}, \underline{10^9}]$

Array size? 10^9

- lots of space

S.

Hashmap.



A[7]

string

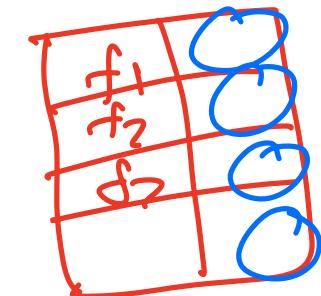
⇒ Hashmap ~~<int, bool>~~ isRoomAvailable;

isRoomAvailable [f_1] = true;

[f_2] = true

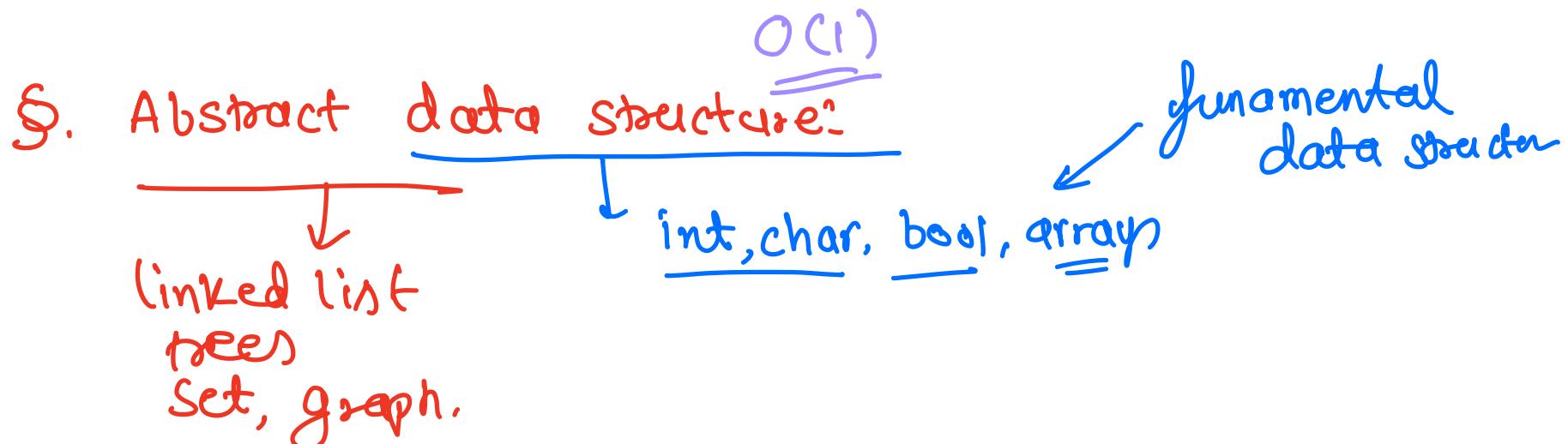
value

Keys



If we have 1000 rooms
hashmap size would be
1000

T.C. of inserting any element in hashmap.

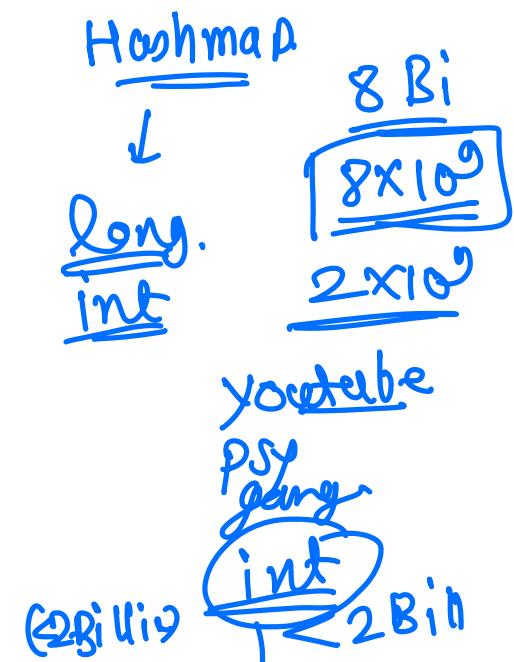


Q. Store population of each country

India	1.4 B
China	1.38 B
:	:
:	:

Hashmap <key, value> Hashmap Name 2

key : string (Country name)
value : int



v

Q. No. of states in the country.

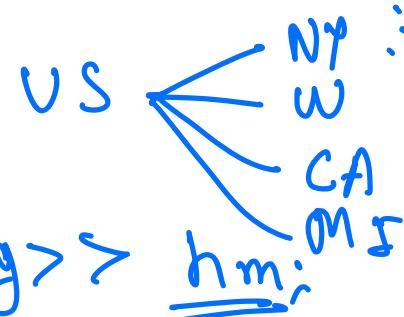
HashMap < String, int >
↓
Country ↑
states

Q. for every country, names of all the states.

Key : Country name

value: String array

HashMap < String, list<String> > hm;



Q For every country, store population of each state as well.



Obsⁿ: value: could be any object

key : are of primitive data type

int / string / float / char
[-2, -1, 0, 1] / double / bool
6. To be unique
[0, 1]

→ keys have to be unique

⊕ intern

Hotel (size, 1000)

⊗ total room NO:

⊗ total available room

⊗ check-in process?

- ① find available room
- ② mark it unavailable

⊗ check-out? → ① we mark the room available

⊗ Delete room.

⊗ HashMap <key, value>

Array A;
A.size()

- ① size ()
- ② insert (key, value)
- ③ delete (key)
- ④ search (key) ← fetch
- ⑤ update (key, value)

$O(1)$ $O(1)$ $O(1)$ $O(1)$ $O(1)$
 $SOL \Rightarrow ?$ on average

* Hashset

data structure

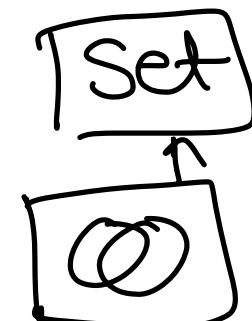
(Set)

Just store unique elements

Hashset <key>

Country names

① size () // is Empty ()



$$S = \{2, 4, 6, 8, \dots\}$$

② insert ()

③ delete ()

④ search ()

⑤ ~~update?~~

(no 'value' to update?)

disadvantage of Hash map / Hash set?

① worst case T.C. to insert $O(N)$

8. Hashing in different programming languages.

pseudo code	Java	C++	Python	JS	C#
-------------	------	-----	--------	----	----

HashMap

HashMap

unordered_map

dict

map

Dictionary

HashSet

HashSet

unordered_set

set

Set

HashSet

Ques

key value

india 1.4 B

"Indiaa" 1.4 B

How to correct this error

delete("india")
insert("india" 1.4 B);

Q. Find frequency of numbers!

Given N elements & Q queries, for each query find frequency of the element in the array.

$1 \leq N \leq 10^5$, $1 \leq Q \leq 10^5$, $1 \leq A[i] \leq 10^9$

ex. $A = [4, 6, 4, 9, 6, 9, 10, 2, 3, 3, 1, 9]$

$Q = 4$
num

freq.

Brute force:
T.C. = $O(Q^*N)$
S.C. = $O(1)$

4
9
11
3

2
3
0
2

- - -

Optimal solⁿ:

① Store the freq. of each element

// array [10⁹] initian to 0
↑ fill it
too much memory

② Hashmap <key, value>

element
(int) ↑ its frequency
(int) ↑

Hashmap <int, int>

void frequencyQuery (A[], Q) {

| Hashmap <int, int> countOf :

A = [4, 5, 4, 3, 5]
↑ ↓

countOf [4, 2]
[5, 2]
[3, 1]

```

int N = A.size()
int qs = Q.size()

for (int i=0; i<N; i++) {
    if (countof.search(A[i]) == true) {
        countof[A[i]]++;
    } else {
        countof.insert(A[i], 1);
    }
}

for (int i = 0; i < qs; i++) {
    num = Q[i];
    if (countof.search(num) == true) {
        print(countof[num]);
    } else {
        print("0");
    }
}

```

T.C. = $(N+Q)$

} }

Q. Given N elements, find the first
non-repeating element.

S.C. = O(N)
all elements are unique
O(1) all elements are same

Ex.

$$A = [1, 2, 3, 4, 5, 6], \text{ ans} = 1$$

$$A = [\underline{4}, \underline{\cancel{3}}, \underline{3}, \underline{2}, \cancel{5}, 6, \underline{4}, \cancel{5}], \text{ ans} = 2$$

$$A = [\cancel{2}, \cancel{6}, \cancel{8}, 4, 7, \cancel{2}, 9], \text{ ans} = 6$$

$$A = [1, 2, 1, 3, 3, 2], \text{ (not a valid input)}$$

ideas: ① Store the frequency

10:50

② Find the element which has frequency == 1.

① insert the elements in the Hashmap O(N)

(I)

② iterate in hashmap and find if the element's frequency is 1. X

$$A = [1, 2, 3, 1, 2, 5]$$

HashMap [[1, 2] [2, 2], [5, 1] [3, 1]]

objⁿ: elements are not stored in order of keys.

$O(N)$

(II)

① iterate over array and check the frequency

T.C. = $O(N)$

S.C. = $O(N)$

Q: Given N elements, find the count of distinct elements.)

ex:

$$A = [1, 2, 3, 4, 5], \text{ ans} = 5$$

~~Not unique~~

$$A = [1, 2, 2, 3, 1, 3] \text{ ans} = 3$$

$$A = [3, 3, 3] \text{ ans} = 1$$

$A = [1, 1, 2, 3, 2, 4]$ ans = 4

idea: ① use hashset
② return hashset.size()

Hashset<int> hs;

```
for (int i=0; i<N; i++) {  
    hs.insert(A[i]);  
}
```

T.C. = O(N)
S.C. = O(N)

return hs.size();

! Can we use hashmap.

✓ we can use hashmap,

hashmap<int, int> hm;

↑
element

↑ Store the frequency

```

for (int i=0; i< N; i++) {
    // if not found
    hm.insert(A[i], 1)
    // else
        hm[A[i]]++;
}

```

Q.4 Given N elements, check if all the elements are distinct or not.

A = [6, 7, 8, 6, 5], ans = No

A = [1, 2, 11, 9, 7, 6], ans = Yes

Idea ① Hashmap to store the frequency

② Use HashSet

ⓐ If HashSet.size() == A.size()

T.C. = O(N), S.C. = O(N)

Q. Given N elements, check if there is a subarray with a sum equal to 0.

ex. $A = [2, 2, 1, -3, 4, 3, 1, -2, -3, 2]$

Ans = Yes

Idea

① Brute force

① for all subarrays $O(N^2)$

② get the sum $O(N)$

③ check sum == 0

T.C. = $O(N^3)$

S.C. = $O(1)$

② Prefix Sum?

① for all subarrays $O(N^2)$

② get the sum $\Rightarrow O(1)$

③ check sum == 0

T.C. = $O(N^2)$

S.C. = $O(N)$

$O(N^2)$

③ Sliding window

④ all the subarrays of size 1, 2, 3,

⑤ check sum == 0. $O(1)$

T.C. = $O(N^2)$

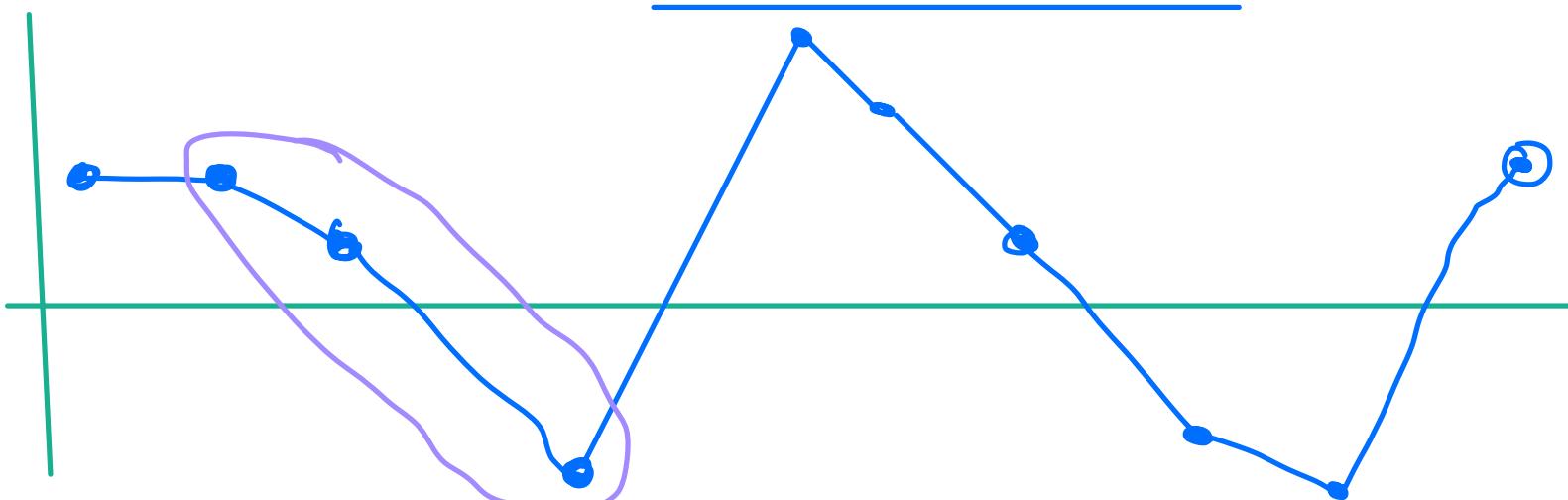
S.C. = $O(1)$

④ optimal soln

① atleast one negative or all 0.

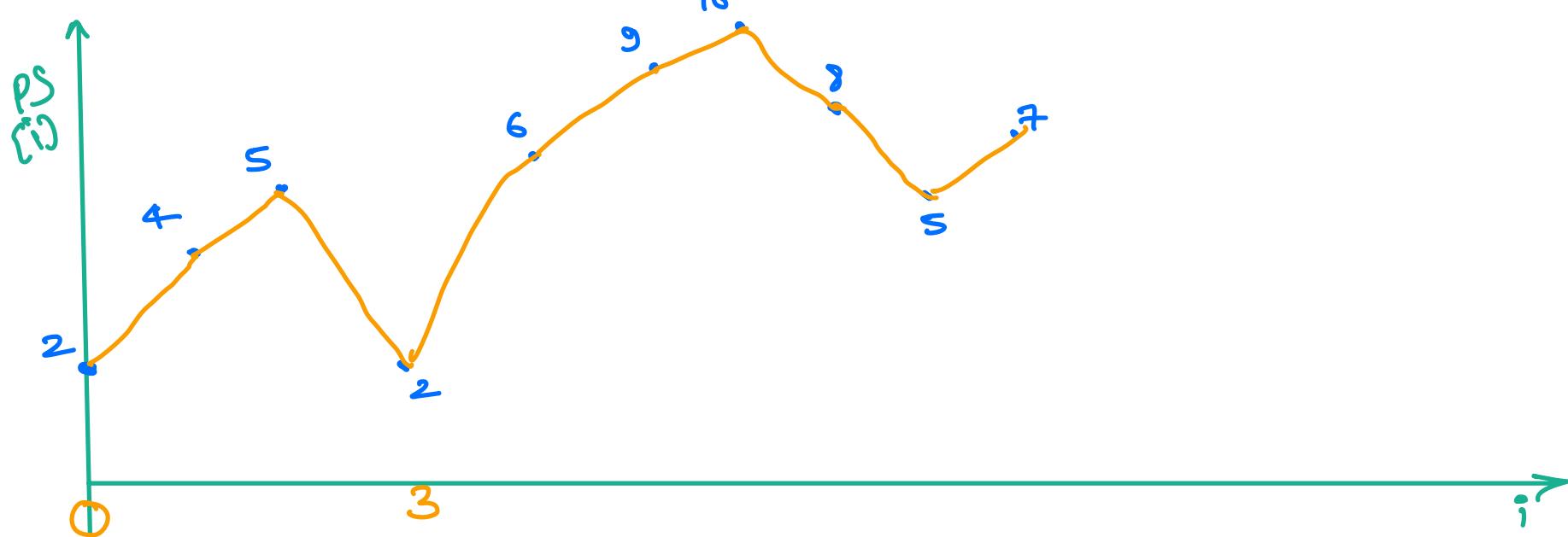
② we can not check all the subarray
 $O(N^2)$

$$A = [2, 2, 1, -3, 4, 3, 1, -2, -3, 2]$$



$PS = [2, 4, 5, 2, 6, 9, 10, 8, 5, 7]$

Stack
area



$$PS[0] == PS[3]$$



Obsⁿ Sum of elements at index (1, 2, 3) = 0.

$$PS[0] = A[0]$$

①

$$PS[3] = A[0] + A[1] + A[2] + A[3] \quad \textcircled{2}$$

$$\Rightarrow A[0] = A[\cancel{0}] + A[1] + A[2] + A[3]$$

$$\Rightarrow O = A[1] + A[2] + A[3]$$

$$\Rightarrow PS[i] = A[0] + A[1] + \dots + \underline{A[i]} \quad \textcircled{3}$$

$$(j > i) \quad PS[j] = A[0] + A[1] + \dots \underline{A[i]} + \dots \underline{A[j]} \quad \textcircled{4}$$

$$\Rightarrow PS[i] = PS[j]$$

\textcircled{3} & \textcircled{4}

$$\cancel{A[0] + A[1] + \dots + A[i]} = \cancel{A[0] + A[1] + \dots + A[i]} + A[i+1] + \dots + A[j]$$

$$\Rightarrow O = A[i+1] + A[i+2] + \dots + A[j]$$

\Rightarrow Sum of elements from index $(i+1)$ to j is zero.

idea

- ① create the prefix sum
- ② If repeating entries in prefixsum
then ans is Yes
else ans is No.

bool checkSubarraySumZero (A) {

 int N = A.size();
 // create prefix sum

 vector<int> PS(N, 0);

 hashset<int> HS;

 PS[0] = A[0]

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

 HS.insert(PS[0]);

```

for (i=1 ; i<N; i++) {
    PS[i] = PS[i-1] + A[i]    T.C. = O(N)
    if (PS[i]==0) return true;
}

hs.insert(PS[1])
}
gt (hs.size() != PS.size());
}

```

$$A = [0 \ 0 \ 0] \quad \swarrow$$

$$A = [1 \ 2 \ 3 \ -6 \ 5 \ 4] \quad \text{ans} = \text{Yes}$$

$$PS = [1 \ 3 \ 6 \ 0 \ 5 \ 9]$$

if any value is zero

$$A = [0, 1, 2, 3]$$

$$PS. = [0 \ 1 \ 3 \ 6]$$

$A = [1, 2, 3, -6, 5, 4, \underline{-9}]$

$PS = [1, 2, 6, \underline{0}, \underline{5}, \underline{9}, \underline{0}]$

Contest Sol 3pm
Contest 2

