

Lecture ÷ Hashing - 1

Agenda

- Introduction
- Internal working [Interview perspective]
- find minimum pairs
- Longest subarray sum = 0
- Length of the longest consecutive subsequence

Quote
सुविचार

÷ कृपया अपने असायिनमेंट्स / psp 80% से
अपट ही रखें। अब तो होमवर्क प्रोबलम्स भी
psp का हिस्सा नहीं हैं, अब बच्चों की
जान लौगा क्या ???

Introduction

arr[] = [2 3 5 5 2 4 2]

freq of each el?

freq[] = [⁰ 0 ¹ 0 ² 2 ³ 1 ⁴ 1 ⁵ 2]

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

→ i = array el
freq[i] = freq.

If the range in array is 10^9 —

freq[$10^9 + 1$] → No. we cannot create an array
of size 10^9 . [MLE]

Memory limit exceeded

usually / generally, arr[10^6] arr[10^7]

* arr[] = [2 3 10^8 10^9 1]

frequency of each el:- freq[$10^9 + 1$] ✗

Map<Integer, Integer> freq = new HashMap<>();

freq.put(10^9 , 1) ✓

☆☆☆

Internals of hashing [Interview perspective]

Hashmap $\xrightarrow{\text{internal}}$ array.

Hashfunction

$$f(x) = x^2 + x + 3 \dots$$

$$x = 2$$

$$f(2) = 4 + 2 + 3 = 9$$

$$f(3) = 9 + 3 + 3 = 15$$

Hashmap:

arr [17]; — Developer

Given arr[] = [10 20 30 27]

Hashfuncⁿ :- $f(x) = x \% 17$ — Developer

$$10 \longrightarrow 10 \% 17 = 10 \longrightarrow \text{idx of array.}$$

$$20 \longrightarrow 20 \% 17 = 3 \longrightarrow \text{idx of array}$$

$$30 \longrightarrow 30 \% 17 = 13 \longrightarrow "$$

$$27 \longrightarrow 27 \% 17 = 10 \longrightarrow "$$

arr[] = [$10^9 + 3$, $10^8 + 2$, $10^7 + 1$]

↓

bucket[10]

hashfunction = $x \% 10$ [0-9]

$$10^9 + 3 \longrightarrow (10^9 + 3) \% 10 = 3 \longrightarrow \text{idx}$$

$$10^8 + 2 \longrightarrow (10^8 + 2) \% 10 = 2$$

$$10^7 + 1 \longrightarrow (10^7 + 1) \% 10 = 1$$

arr[] = [10 20 30 27]

Hashfuncⁿ :- $f(x) = x \% 17$ → Developer

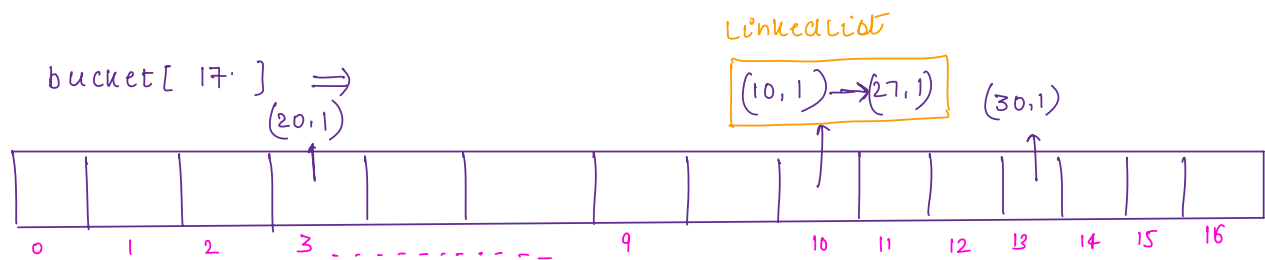
10 → $10 \% 17 = 10$ → idx of array.

20 → $20 \% 17 = 3$ → idx of array

30 → $30 \% 17 = 13$ → "

27 → $27 \% 17 = 10$ → "

$$\left[\begin{array}{l} 10 - 1 \\ 20 - 1 \\ 30 - 1 \\ 27 - 1 \end{array} \right]$$



LinkedList<Node> bucket[17]; → Hashmap
internal

```

class Node {
    int key;
    int value;
}

```

20 → hashfunc val of 20 ⇒ idx ⇒ insert in bucket
└ O(1)

30 → O(1) insert

27 → hf of 27 → idx = 10 → insert at 10th idx in bucket

Worst case scenario ⇒ insertion in map = O(n) time

arr[] = [10 20 30 27]

Hashfuncⁿ :- $f(x) = x \% 17$ \rightarrow Developer

10 \rightarrow $10 \% 17 = 10 \rightarrow$ idx of array.

20 \rightarrow $20 \% 17 = 3 \rightarrow$ idx of array

30 \rightarrow $30 \% 17 = 13 \rightarrow$ "

27 \rightarrow $27 \% 17 = 10 \rightarrow$ "

collision

Good hashmap

\rightarrow Less no of collisions

\rightarrow Hashfunction \Rightarrow Good

Is it possible to avoid collision? [NO] \rightarrow Reduce collisions

range :- 1 ---- 10^9

bucket[] = 10^6

Pigeon hole principle

n holes

n+1 pigeons

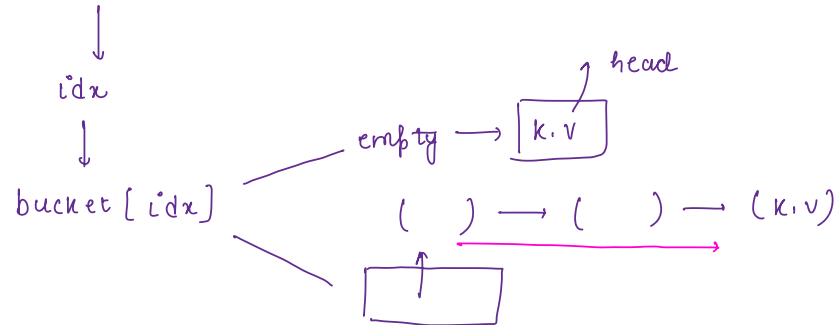
\Rightarrow

There will be one hole which will have 2 pigeons

Nice to know

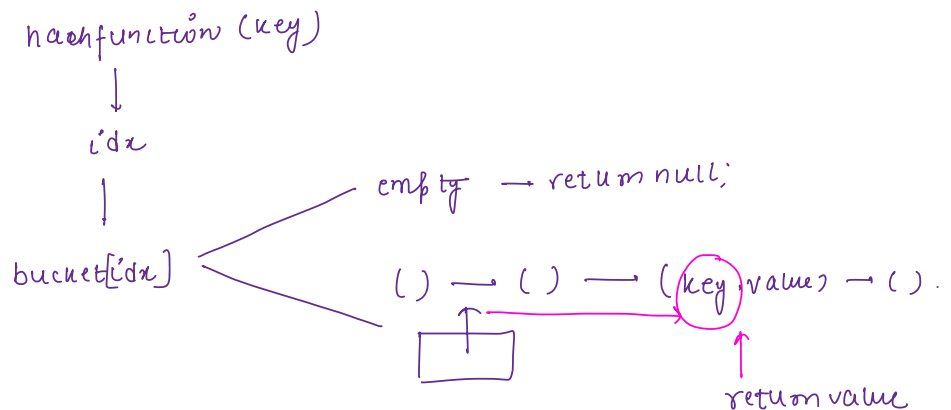
`Map<Integer, Integer> map = new HashMap<>();`

insert / put :-
(k, v)



Avg. TC: $O(1)$

search / get :-
(key)



Avg TC: $O(1)$

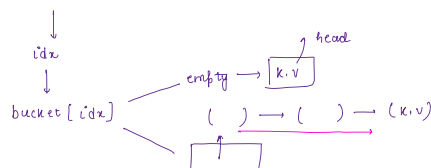
* string

`Map<String, Boolean> map;`

Insert :- $O(\text{len of string})$

(k, v) \rightarrow `hashfunction(key)` \Rightarrow "ayush"
 $O(k) = O(\text{len})$

TC: $O(\text{len})$



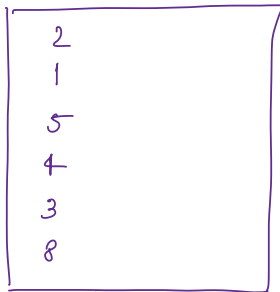
Initialisation

map := HashMap<_____, _____> map;
set := HashSet<_____> set;

} — unsorted map

arr[] = [2 8 1 3 5 4]

map / set

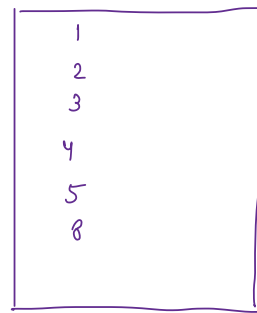


unsorted / unsorted

sorted map

Map<_____, _____> map =

new TreeMap<>();



Balanced BST. (Trees)

TC := $O(\log n)$

Conclude:

freq[i] = normal array

update = $O(1)$

Issue := range $[10^9]$

HashMap



freq[] \rightarrow LinkedList<Node>[]

TC: $O(1)$

↑
integer / double /
boolean

strings

↓
 $O(\text{len of string})$

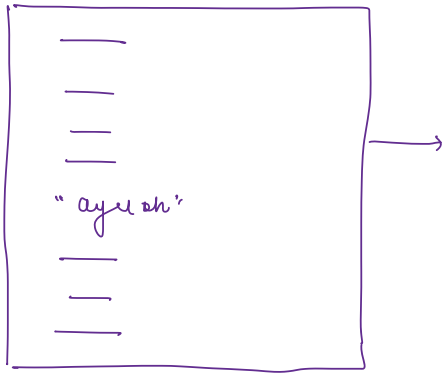
→ for a good hashmap, good hashfunction, less no. of collisions.

Break 8:42 AM

arr[10⁵] = [_ _ _ _ "ayush" _ _ _ _]

$$1 \leq \text{arr}[i] \leq 100$$

search(ayush) = 10^5 . $\neq O(n)$



search | get Caysuohs.

hash(ayush) $\rightarrow O(K)$

$$\left. \begin{array}{l} \downarrow \\ \text{idx} \\ \downarrow \end{array} \right\} \text{ol(1)}$$

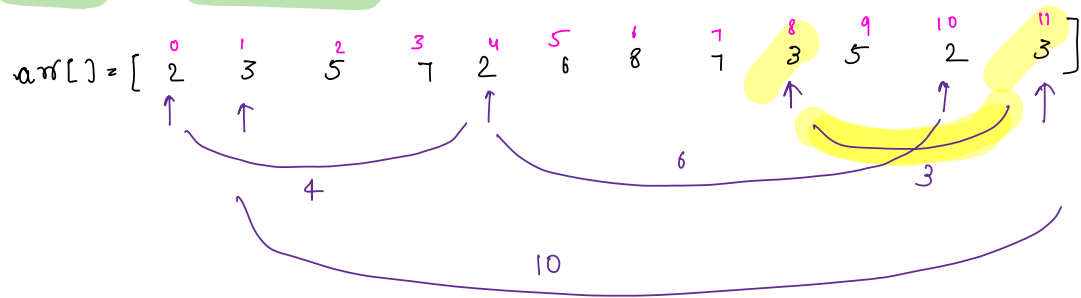
string str = "aaabbbcc"

$$\downarrow \left[\begin{array}{r} a : 3 \\ \hline b : 3 \\ c : 2 \end{array} \right]$$
$$\rightarrow \left[\begin{array}{l} \text{map} \\ \textcircled{3} : \langle a, b \rangle \\ 2 : \langle c \rangle \end{array} \right]$$

$\text{map}(\text{Char}, \text{Integer}) \rightsquigarrow \text{map}(\text{Integer}, \text{List}(\text{Char}))$

Q4: Given integer array, find any pair (i, j) such that ~~***~~

$i \neq j$, $A[i] = A[j]$ and $|i - j|$ is minimum.

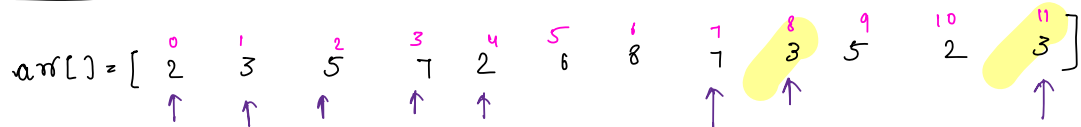


Brute force: Try all the pairs

TC: $O(n^2)$

SC: $O(1)$

Approach 2



map:

key (arr el)

value

2

~~0~~ ~~4~~ 10

3

~~1~~ ~~8~~ 11

5

~~2~~ 9

7

~~3~~ 7

6

5

8

6

ans. = ~~4~~ 3

Q2: Given $arr[n]$, find length of longest subarray with $sum = 0$

$arr[] = [\overset{0}{1} \quad \overset{1}{-2} \quad \overset{2}{2} \quad \overset{3}{-2} \quad \overset{4}{-3} \quad \overset{5}{-1} \quad \overset{6}{4} \quad \overset{7}{7}]$

$pf[] = [1 \quad -1 \quad 1 \quad -1 \quad -4 \quad -5 \quad -1 \quad 6]$

$pf[i] = \text{subarray sum: } 0 \text{ to } i$

$\rightarrow pf[i] == 0 \quad [\text{subarray}(0, i) \xRightarrow{\text{sum } 0}]$

\uparrow
 $len = i + 1$

$\rightarrow pf[e] - pf[s-1] \rightarrow \text{subarray}(s, e)$

$pf[e] - pf[s-1] = 0$

$pf[e] = pf[s-1]$

$arr[] = [\overset{0}{1} \quad \overset{1}{-2} \quad \overset{2}{2} \quad \overset{3}{-2} \quad \overset{4}{-3} \quad \overset{5}{-1} \quad \overset{6}{4} \quad \overset{7}{-1}]$
 $pf[] = [1 \quad -1 \quad 1 \quad -1 \quad -4 \quad -5 \quad -1 \quad -2]$ } $O(n)$

$O(n)$ {

key ($pf[i]$)	value	ans [idx - min idx]
1	0	
-1	1	$\neq 5$ [return 5]
-4	4	
-5	5	
-2	7	

TC: $O(n)$

SC: $O(n)$

Q Longest consecutive subsequence [order does not matter]

arr[] = [100 4 3 6 10 20 11 5 101]

Consecutive subsequences: 100 101

10 11

5 6

longest cs \Leftarrow 3 4 5 6

Approach 1 sort array —

arr[] = [3 4 5 6 10 11 20 100 101]

$A[i] + 1 = A[i+1]$, continue sequence

else; stop.

update your ans

start with another seq

TC: $O(n \log n)$

SC: $O(1)$

Approach 2

arr[] = [100 4 3 6 10 20 11 5 101]

min = 3

max = 101

hashset

100
4
3
6
10
20
11
5
101

3 4 5 6 7 8 9 ... 101
↑

Traverse from min to max (el) }

if (el is in hashset) {

 check from el+1 is hashset

 keep updating your ans

}

TC: $O(\text{max} - \text{min}) + O(n)$

SC: $O(n)$

What if range of el is $[1 - 10^9]$

Approach 3

arr[] = [⁰100 ¹4 ²3 ³6 ⁴10 ⁵20 ⁶11 ⁷5 ⁸101]

0	is 99 present in array? 100 can be starting bt of a subsequence 100 101 \rightarrow ans=2
4	3 present? Yes
3	2 present? 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 ans=4
6	5 present? yes
10	9 present? 10 can be starting bt of subsequence 10 \rightarrow 11 ans=4
20	19 present? 19 ans=4
11	10 present. Yes
5	4 present? Yes
101	100 present? Yes

```

int longestConsecutive(int[] arr) {
    Set<Integer> set = new HashSet<>();
    for (int el : arr) {
        set.add(el);
    }

    int ans = -1;
    for (i=0; i<n; i++) {
        if (set.contains(A[i]-1)) {
            continue;
        } else {
            future-el = A[i] + 1;
            cnt = 1;
            while (future-el is in hashset) {
                cnt++;
                future-el++;
            }
            ans = max(ans, cnt);
        }
    }
    return ans;
}

```

$O(n)$

TC: $O(n)$

SC: $O(n)$

$arr[] = [\overset{0}{100} \quad \overset{1}{4} \quad \overset{2}{3} \quad \overset{3}{6} \quad \overset{4}{10} \quad \overset{5}{20} \quad \overset{6}{11} \quad \overset{7}{5} \quad \overset{8}{101}]$

100 \longrightarrow check for 99 in set $O(1)$

while ()

100, 101 \longrightarrow ~~4~~ el.

4 \longrightarrow check for 3 $O(1)$

3 \longrightarrow 2

3, 4, 5, 6 \longrightarrow ~~4~~ el

6 \longrightarrow 5 present.

$arr[] = [1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6]$

1 \longrightarrow 0 present

1, 2, 3, 4, 5, 6 $\longrightarrow O(n)$

2 \longrightarrow 1 present?

$O(n) \times$

3 \longrightarrow

$O(n) \times$

4 \longrightarrow

$O(n) \times$

Thankyou 😊

Doubt:

[3, 8, 11, 12, 19, 105, 1001]
 ↑ ↑ ↑ ↑
 1 step 10 step
1 step 10 step
 arr[7]

[1 3 5 9 10 ... 91 [2 4 6]]
 ↑ ↑
 2

```

int longestConsecutive(int[] arr) {
    Set<Integer> set = new HashSet<>();
    for (int el : arr) {
        set.add(el);
    }

    int ans = -1;
    for (int i = 0; i < n; i++) {
        if (set.contains(A[i] - 1)) {
            continue;
        } else {
            future-el = A[i] + 1;
            cnt = 1;
            while (future-el is in hashset) {
                cnt++;
                future-el++;
            }
            ans = max(ans, cnt);
        }
    }
    return ans;
}
  
```

$O(n)$