Monday, 15 July 2024
8:44PM

Homework

Deadline

Friday → 19

↳ 4Days

20+ B.S

→ 20+ Hashing

35

Hashcode

↳ Integer → Integer

Double

String S, n

hashcode ↳ $S_0 \times 31^{n-1} + S_1 \times 31^{n-2} + S_2 \times 31^{n-3} + \dots + S_{n-1} \times 31^0$ ✓

↳ It taken O(n) to calculate

Map → Key, Value

str → a b a c a b a a c c.

freq of each alphabet ??

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

↑ char ↑ int

Map

a	3 5
b	3
c	3

→ O(1) ??

map.get('c')

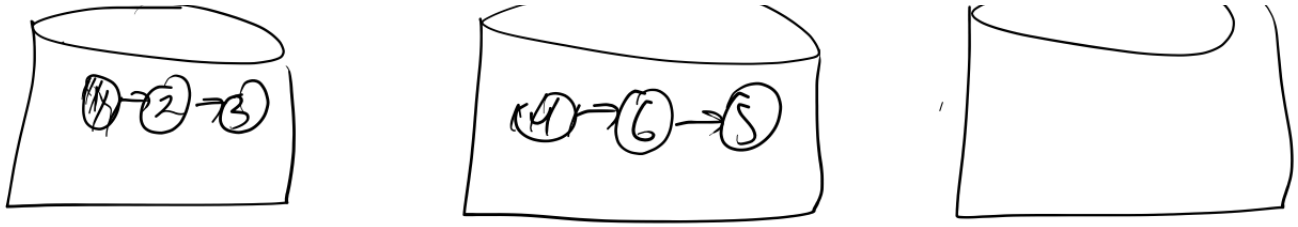
↑ O(LF)

O(1)

Hashset → HashMap<Integer, Boolean>

↳ Not a key value store

↳ just stores keys



Remove a node from LL

① → N → null

head = head.next

N → N1

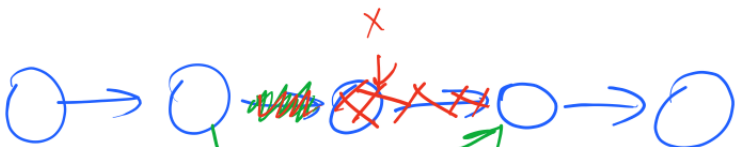
② Key

X → N → Y

prev

next

prev.next = next



Missing Numbers

A1 → 10: 203 204 205 206 207 208 203 204 205 206

A2 → 13: 203 204 204 205 206 207 205 208 203 206 205 206 204

203 - 2 ✓
204 - 2
205 - 2
206 - 2
207 - 1
208 - 1

All elements that present only in A2, not in A1

Or

having different counts in A1 and A2

203 - 1 + 1 = 2
204 - 1 + 1 = 2 + 1 = 3
205 - 1 + 1 = 2 + 1 = 3
206 - 1 + 1 = 2 + 1 = 3
207 - 1
208 - 1

→

203 - 2
204 - 3
205 - 3
206 - 3
207 - 1
208 - 1

204
205
206

Steps

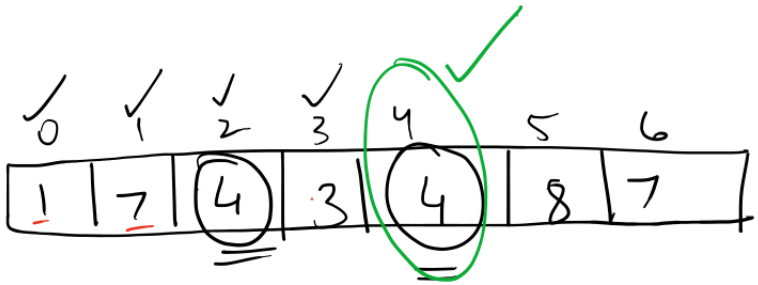
↳ Make 2 Hashmaps storing the frequency of each element

↳ Iterate over the Hashmap of A2

and identify the missing numbers.

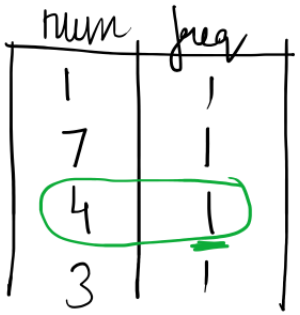
First element to occur (k) times

72
1743487



1-1 x

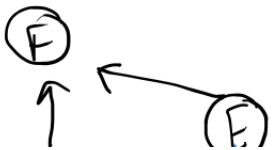
1)



1+1 == k

Manager & Employees

6
AC
BC
CF
DE
EF
FF

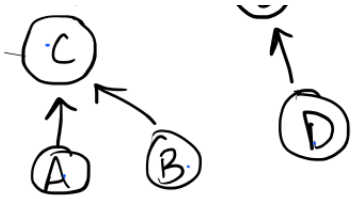


A : 0
B : 0
C : 2

→ Who are the reportees?

A: 0
B: 0
C: A, B 2
D: 0
E: D 1
F: C, E

2 + 1 + 2 = 5



D: 0
E: 1
F: 5
G: 6
H: 7

Hint : Recursion

1) KeySet() → Returns a set of all the keys

2) entrySet() → Returns all the map entries

Entry → Key
↳ value

```
for(Map.Entry<String,String> entry : emp.entrySet()){
        String key = entry.getKey();
        String value = entry.getValue();
```

```
}
```