Q: You are given a list of size n, where all the elements of the list are integers. Every element in the list is present twice except for one special element. Find the special element.

$n \leq 10^6$

Ex [1, 2, 1, 3, 2]

ans → 3

→ we can't use any extra data structure

→ you can't use any internal functions

Brute force

[1, 2, 1, 3, 2] ⟶ $O(n^2)$

for ——

for ——

$n \le 10^6$

$10^8 < 10^{12}$

≤ 1 sec

↳ all elements are present in a pair. except one element. Can we any how eliminate the elements present in pair.

xor

$a \wedge a = 0$

$0 \wedge y = y$

We can iterate over the list, & bring out the xor of all the elements. The final value will be our ans.

[1, 2, 1, 3, 2]

x = 1 ^ 2 ^ 1 ^ 3 ^ 2 ⇒ 3 ans

x = x ^ 1

x = x ^ 2

x = x ^ 1

x = x ^ 3

x = x ^ 2

→ dict { <key>: value,
<key>: value,
}

→ Hashing

avg case

insertion
deletion
search
→ O(1) const time

worst case

insertion
deletion
search
→ O(n)

probability of worst case is extremely less.

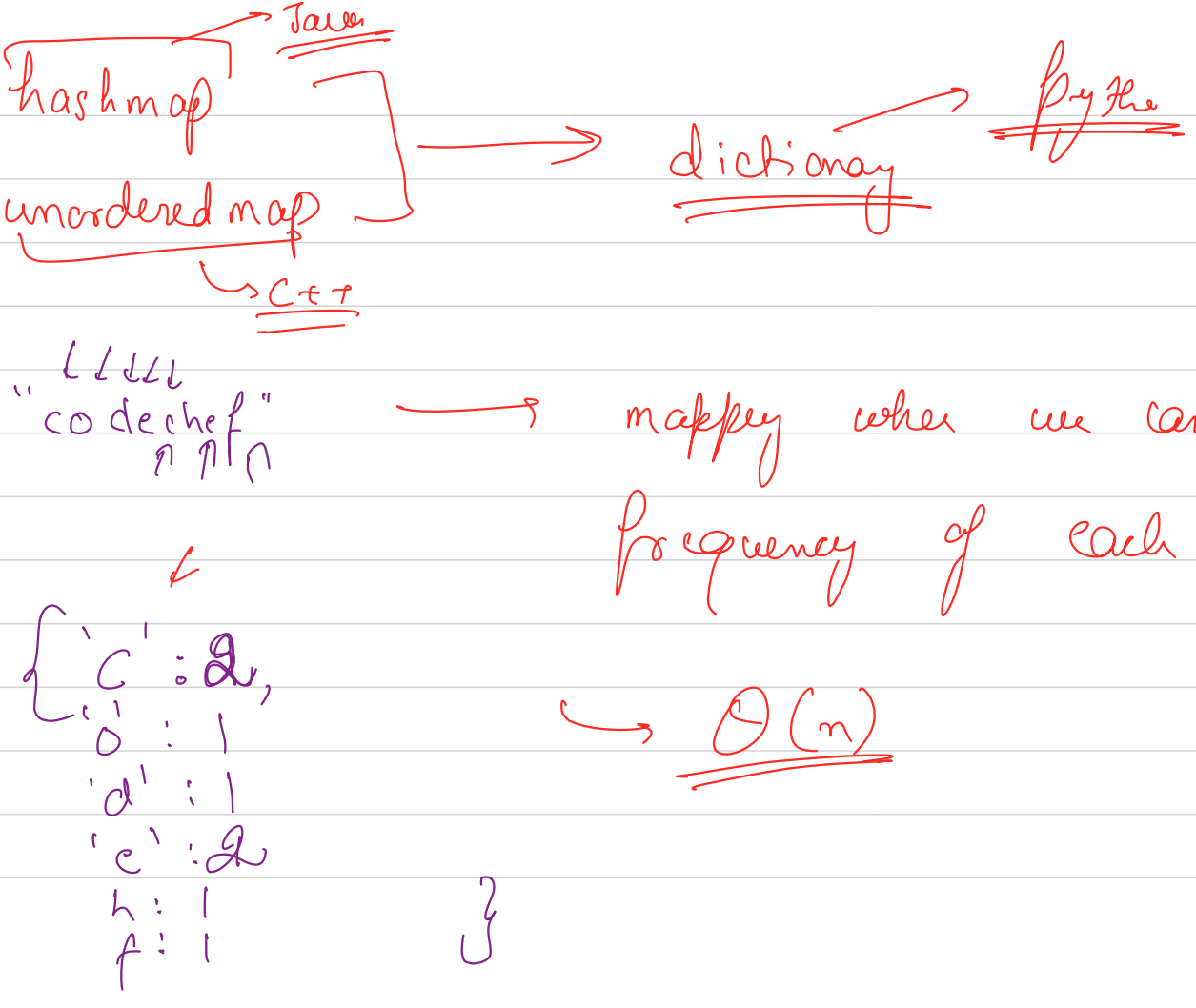
Dictionary

Q⇒ You are given a string of length n. Find all the characters which are non-repeating

$n \leq 10^7$

Ex "unacademy" → ans → u n c d e m y

"codechef" → o d h f

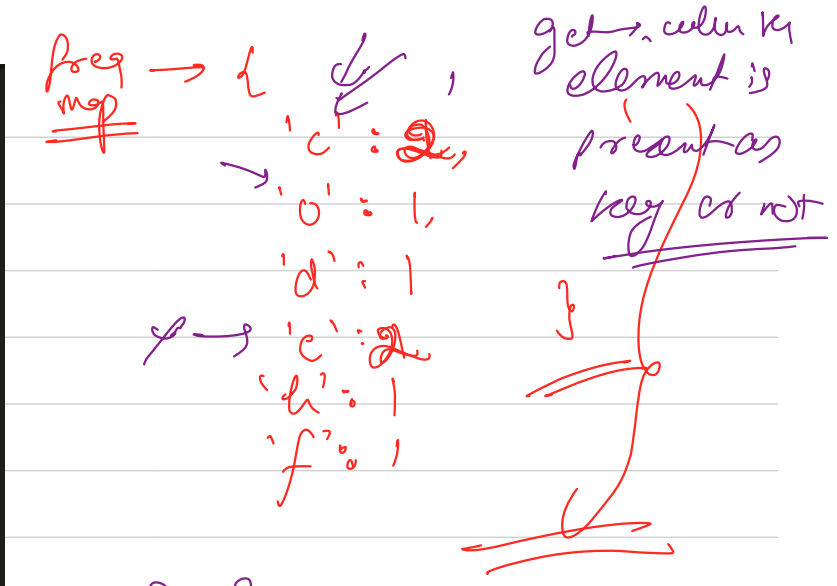

Java
hashmap
unordered map
C++
dictionary
by the
"codechef"
mappy when we can store
frequency of each character.
{ 'c' : 2,
'o' : 1
'd' : 1
'e' : 2
h : 1
f : 1 }
O(n)

```python
def getUniqueCharacters(string):
    freq_map = {}
    for i in range(0, len(string)):
        if(freq_map.get(string[i]) == None):
            freq_map[string[i]] = 1
        else:
            freq_map[string[i]] += 1

    result = []
    for k in freq_map.keys():
        if(freq_map[k] == 1):
            result.append(k)

    return(tuple(result))


string = "codechef"
result = getUniqueCharacters(string)
print(result)
```



"codechef"

1 dimensional list → [ ]

2 dimensional list → list of lists

```
[ [1, 2, 3, 4],
  [5, 6, 7, 8],
  [9, 10, 1, 2] ]
```

→ 2 dimensional list

```python
n = int(input())
m = int(input())

for i in range(0, n):
    y.append([]) # appending the ith row
    for j in range(0, m):
        x = int(input())
        y[i].append(x)

print(y)

print(y[0][1]) # accessing 1st column of the 0th row
```