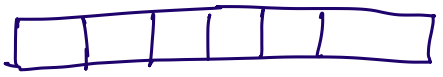
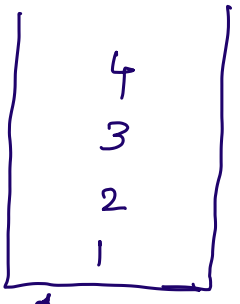
① Contains duplicate.

Given →  → array

output → true / false → it array contains duplicates.

array

put in hashset



hashset

Removes duplicate

compare length of array with. with size of hashset.

size 6

size 4

means something was removed so there was duplicate in array.

② Valid Anagram –

Given → S_1, S_2

Out → Anagrams or not → true/false

create freq. Map of each string

raacar

r	2
a	2
c	2
e	1

carrae.

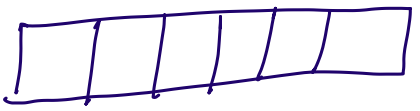
c	2
a	2
r	2
e	1

compare

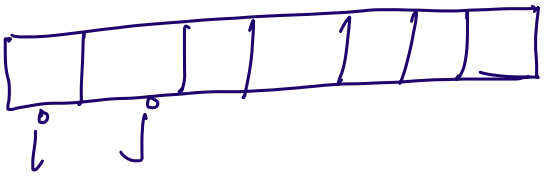
they are same → true

otherwise → false.

③ Two Sum -

Given →  & targetSum.

output → indices
when makes targetsum.



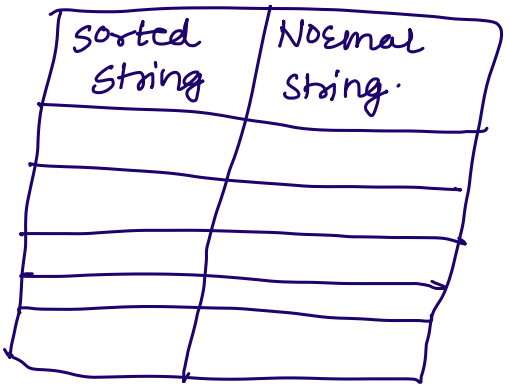
check if arr[i] == arr[j]
if yes return arr{i, j}

for every i
check j.

for (i=0 → i<n-1)
for (j=0 → j<n-1)

④ Group anagrams.

We take every string & sort it.



Hashmap

if string matches sorted string

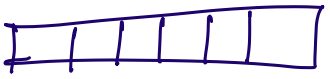
add the normal version

We match every string by sorting it

so that we could find anagrams.

Return all values as list.

⑤ Top K frequent elements.

Given →  Top → 2.

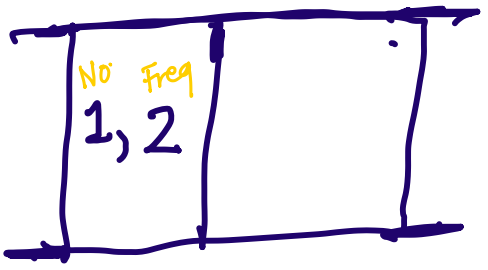
arr = [1, 1, 2, 2, 2, 2, 3, 33]

No	freq.
1	2
2	4
3	3

Hashmap

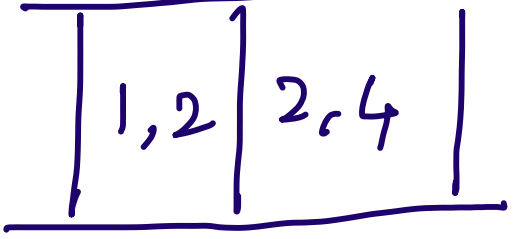
→ min heap of size (Top)

size-2



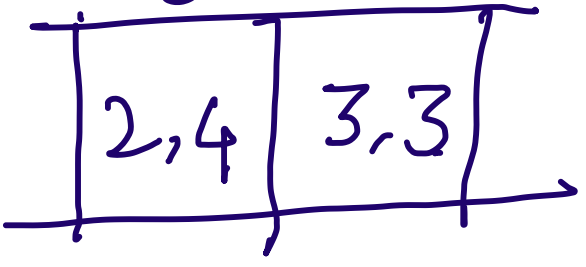
PQ

size-2

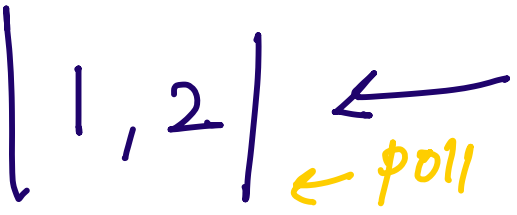


PQ

size-2



PQ



← poll

(6) encode & decode strings -

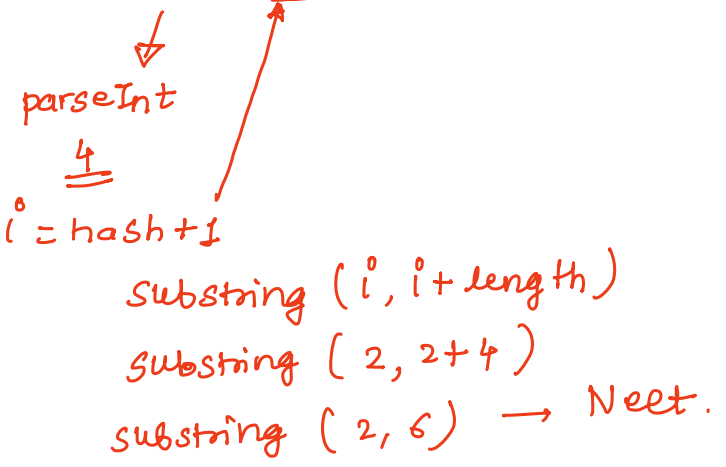
[Neet, code, love, you].

encode. → length + # + string

4#Neet 4#code 4#love 3#you

decode. →

4#Neet 4#code 4#love 3#you

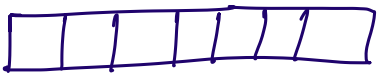


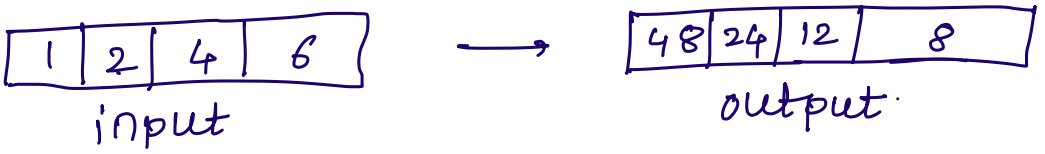
substring (i, i+length)

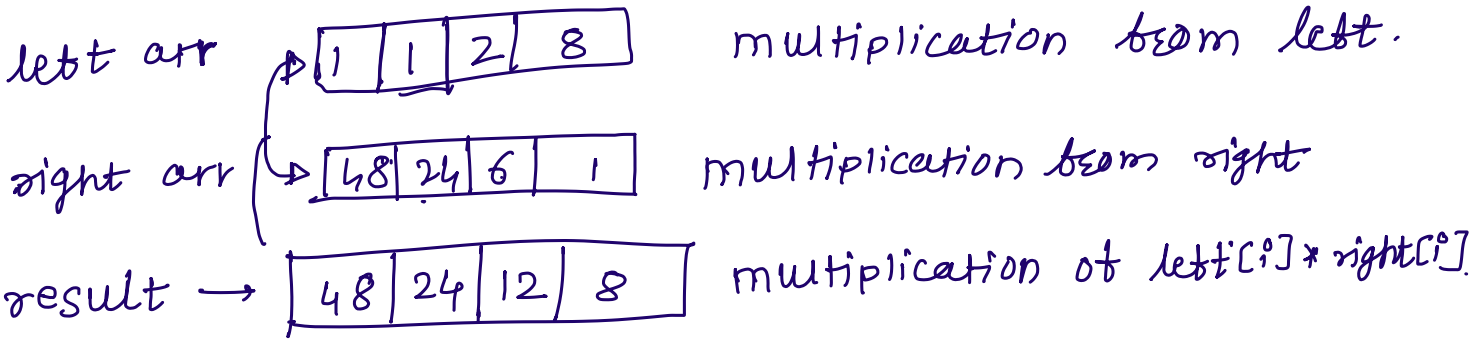
substring (2, 2+4)

substring (2, 6) → Neet.

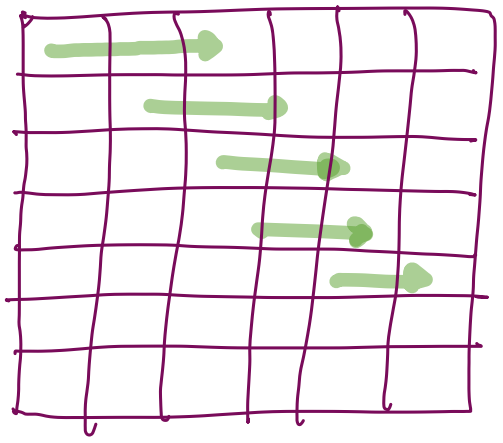
⑦ Product of array except itself.

→ Given → 

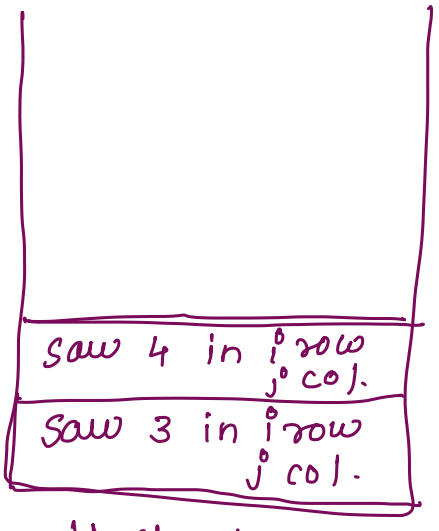




⑧ Valid sudoku



traverse matrix



→ we keep adding strings with No & its row & col.

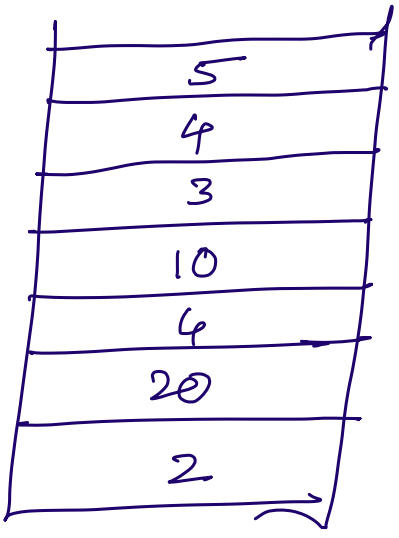
& if we see the same number in same row or same col or even diagonally we return false -

for diagonal we use i/3 & j/3.

⑨ longest consecutive sequence.

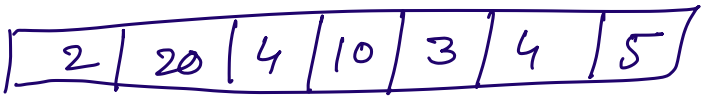
Given →  ← Non sorted

put everything in hashset for constant access



Hashset

Now we traverse array



(2) do we have 3 in hashset

2 → 3 → do we have 4.

2 → 3 → 4 do we have 5

2 → 3 → 4 → 5 do we have 6.

longest till Now would be 2-3-4-5.

we only start counting length when we dont have n-1 number

eg

we got '4' in array

we only start counting if there does not exist 3.

if 3 exist, we would start by '3' in future.