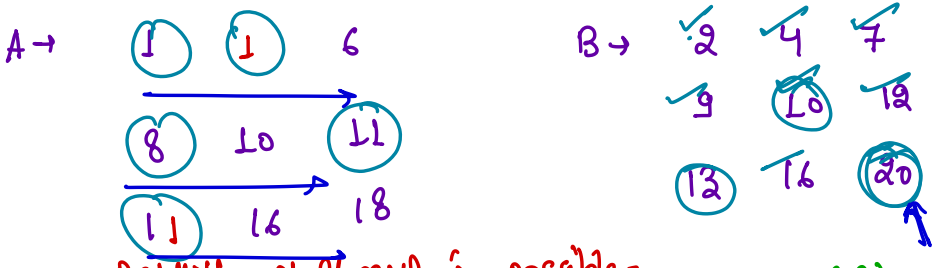
1. Pair with Given Sum in two sorted matrices:

Date: 16th January 2022

1. Pairs With Given Sum In Two Sorted Matrices
2. Smallest Subarray With All Occurrences Of The Most Frequent El...
3. X Of A Kind In A Deck
4. Brick Wall



→ Repetition of Element is possible.

Target = 21

time: $O(n^2)$
space: $O(n^2)$

① Add element of matrices A in Hashmap with freq.

② Travel on matrix B, set first element of pair as val2 = B[i][j], and find val2 in A ? search for target − val2 in Hashmap.

Elements of matrices are randomly arranged.

n^2 space

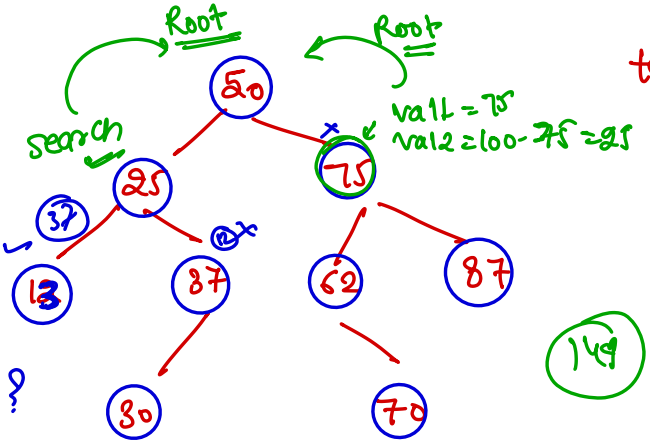
1 → ~~1~~ 2
6 → 1
8 → ~~1~~
10 → 1
11 → ~~1~~ 2
16 → 1
18 → 1

21 − 10 = 11
val1 = 2 21 − 9 = 12
val2 = 21 − 2 = 19
= 21 − 4 = 17
21 − 7 = 14
count = ~~0~~ ~~2~~ ~~2~~ 1

21 − 12 =
21 − 16 = 5

21 − 20 = 1

Given that →



target → 100

print all pairs having sum = 100

25 ↔ 75 → 100

75 ↔ 25

How to avoid these duplicacy ?

149

target = 149

find greatest value -

25 - 75
30 - 70
13 - 87

} val1 < val2 → find

val 75 → 25 ✗

val1 + val2 = target

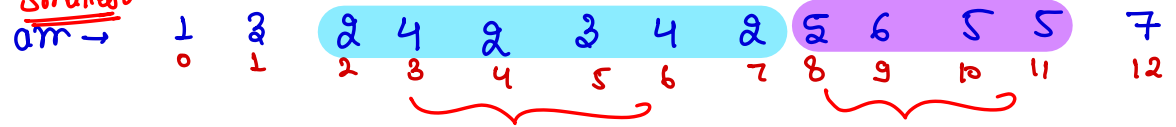
val1 > val2

val2 > val1

val1 == val2 ✗

] → distinct element

Smallest subarray with all occurence of the most frequent element:



Fmap →

1 → 1
3 → 2
2 → 3 } Most frequent Element
4 → 2
5 → 3 } Most frequent Element ||
6 → 1
7 → 1

① Max freq. → H.M. ①
② Starting of that Element → H.M. ②

Output
- Element having max freq.
- Starting Index
- Ending 4

X of a kind in a deck of Cards :

arr → 1 2 3 4 4 3 2 1

Simple arr.

↳ partition → size ≥ 1

partition size ≥ 2

Split in partitions.

(1) partition contains similar Element

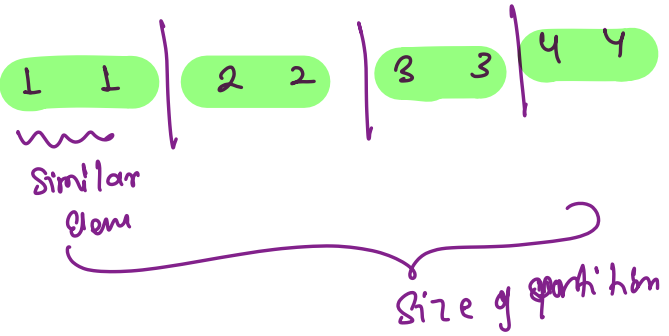
(2) size of all partition should be same } if possible → True, else → False.

1 → 2

2 → 2

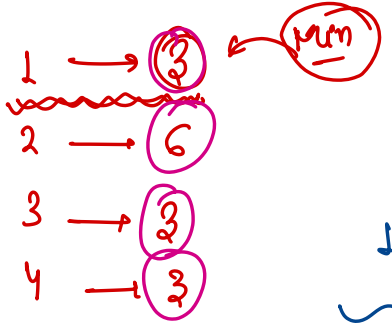
3 → 2

4 → 2



arr → 1 1 2 3 2 1 4 4 2 4 2 3 2 3 2

freq.

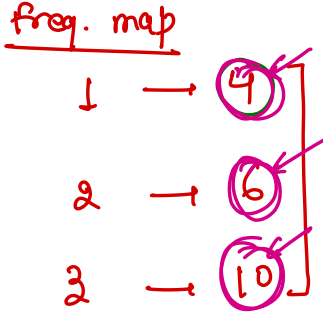


1 1 1 | 2 2 2 | 2 2 2 | 3 3 3 | 4 4 4

① partition have similar type element

② partitions have equal size,

Conclusion → partition size ~~should be~~ smaller freq



→ splitting ?

GCD

if GCD >= 2 → Result → True

otherwise → False.

Steps to solve prob:

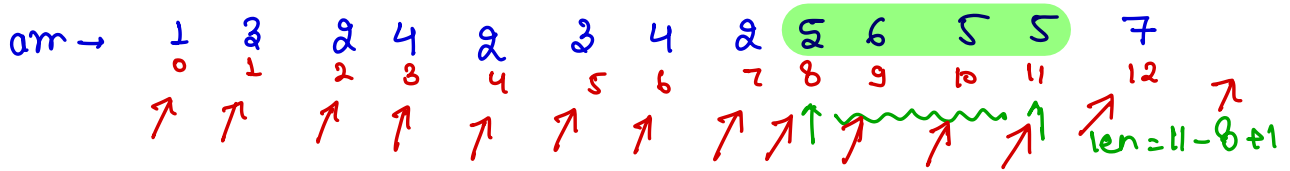
(1) Make freq. map.

(2) Find GCD of frequencies.

freq. 4, 6, 10

GCD → 2.

1 1 | 1 1 | 2 2 | 2 2 | 2 2 | 3 3 | 3 3 | 3 3 | 3 3 | 3 3



fmap

1 → 1
2 - ~~1~~ 2
2 → ~~1~~ ~~2~~ 3
4 → ~~1~~ 2
5 - ~~1~~ ~~2~~ 2
6 - 1
7 → 1

Starting Index

1 → 0
3 → 1
2 → 2
4 → 3
5 → 8
6 → 9
7 → 12

Space → O(n)

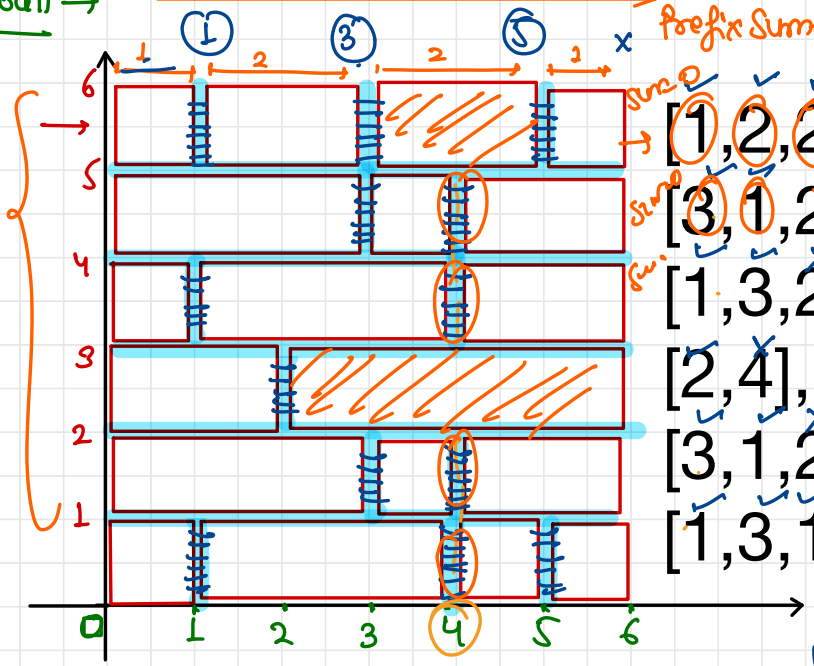
max. freq = ~~1~~ ~~2~~ 2
start = ~~0~~ ~~4~~ 8
end = ~~0~~ ~~4~~ current index 7 11
length = ~~1~~ ~~2~~ (6) 4
↳ [end - start + 1]

(freq < maxfreq)
⇒ Nothing to do

freq == maxfreq
check for length
upgradation.

freq > maxfreq.
change max freq.
start, end, length.

Brick wall →



Prefix Sum

[1,2,2,1],
[3,1,2],
[1,3,2],
[2,4],
[3,1,2],
[1,3,1,1]

Alternate → Max joint

size of brick

point vs freq. coordinate

1 → ✗ ✗ 2
3 → ✗ ✗ 2
5 → ✗ 2
4 → ✗ ✗ ✗ (4)
2 → 1

having max. joint

Size of layer

Max joint → (4)

min bricks cut

line along -y axis than we see joint on x axis

line parallel to y-axis to min cuts in bricks ≡ line parallel to y-axis for max cuts in joints

Max joint = (4)

bricks = size of layer − max joint
= 6 − 4 = (2)