Then we will Cross of left and right So returning maximum we found till now which 24

Hash maps & set

Hashmap are efficient in lookups, insertion & deletion

Properties of hashmap

1) Data is stored key value pair
2) hashmap don't store duplicate key
3) hashmaps are not sorted in any specific order

if problem has keywords like frequency, unique, map, dictionary or fast Lookup, then falls under hashmap or set

Pair Sum unsorted

we need to find a pair whose sum equals target in unsorted array

we can solve this problem by hashmaps

step by step approach

1) we need to find a pair (x, y) sum to target

$$x + y = \text{target}$$

that means $x + y = target$

we can $target - x = y$

2) we need to return index of pair so we can use something like key value pair where value holds the index.

3) while iterating the array. we need to check we need create dictionary which hold value of array and its index

4) we need check target - current element exist in the arr. which means we have already seen y part in array

5) iterate till end of array or till valid pair is found

[-1, 3, 4, 2]   target = 3

dict = {}   target - arr[i] =

i = 0   3-(-1) = 4

4 does'nt exist dict . we need to upate the dict arr[0] and it index.

dict = { -1 : 0 }   target - arr[i]

i = 1   3 - 3 = 0

again 0 does'nt exist in dict so we upate dict with 3 and index as 1

dict = { (-1:0), (3,1) }   3 - 4 = -1

i = 2

So we upate dict correspondingly

dict = {(-1,0), (3,1), (4,2)}

here -1 exist in dict. So we return their index as -1 and current element Index

Verify sudoku board

1) ~~we~~ we have grid 9x9 sudoku board. verify current state of board adhere to the rule of board.

1) each row & column should be unique

2) each sub grid must be unique.

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

step by step approach

1) we need to find each row has unique value at this moment

2) To verify that we can use hashset to find unique values

3) Same for each column we need to maintain a hashed to verify it is adhere to rules of sudoku

4) but to verify each subgrid 3x3 is complex.

5) To do that we are ~~going~~ going to maintain 2D hashset

for subgrid 1

subgrid2

$$\begin{bmatrix} set() & set() & set() \\ set() & set() & set() \\ set() & set() & set() \end{bmatrix}$$

6) How find where to lookup for hashset when we are visiting element in sudoku

7) in a row wise 0-2. belongs to sub grid-1

3-5 belong sub grid-2

6-8 belong sub grid-3

in column wise

0-2 -> subgrid-1, sub grid-2, sub grid-3

3-5 -> sub grid-4, 5, 6

6-8 -> sub grid - 7, 8, 9

8) which when we divide row number by 3 and column number by 3 we arrive the Index to lookup for 2D hashset.

0~~-2~~

row = 9

column = 9

[r/3] [c|3] => [9/3] [9/3] = [3][3]

which means ~~[illegible]~~ at element 9 by 9 need ~~b~~ to be verified 3x3 hashset

Zero Striping

Given mxn matrix. if an element ~~at~~ is zero set it entire row & column as zero

This problem can be solved using hashset

STEP by STEP approch

1) Create two hashset for tracking the column that need to be zero & row also.

2) ~~Kas~~ rowhashset to track column
   ColumnHashset to track row

3) loop the matrix one by one. if element is zero. then add rownumber to rowHashset and column ~~no~~ number to Column Hashset.

4) Then loop the matrix again. now check ~~[illegible]~~ current rowNumber present in hashset or columnNumber present in hashset.

5) if yes the make the current element as zero

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 2 & 3 & 8 & 6 & 7 \\ 1 & 1 & 2 & 3 & 0 & 4 \\ 2 & 5 & 0 & 6 & 7 & 8 \\ 3 & 9 & 10 & 11 & 12 & 13 \end{array}$$

now we have two hashset

rowHashset = { }
col Hashset = { }

after loop the ~~[illegible]~~ matrix we will have like this

rowHashset = {1, 3}
col Hashset = {1, 2}

Then loop the matrix again make the elements zero accordingly