

Then we will ~~as~~ 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 hash map

- 1) Data is stored key value pair
- 2) hashmap don't store duplicate key
- 3) hashmap can not be 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

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

we can $\text{target} - x = y$

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 array which means we have already seen y part in array

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

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

dict = { }
 $i = 0$ target - arr[i] =
 $3 - (-1) = 4$

4 doesn't exist dict. we need to update the dict $\text{arr}[0]$ and its index.

$\text{dict} = \{-1: 0\}$

$i = 1$

target $-\text{arr}[i]$

$$3 - 3 = 0$$

again 0 doesn't exist in dict so we update dict with 3 and index as 1

$\text{dict} = \{(-1: 0), (3, 1)\}$

$$3 - 4 = -1$$

$i = 2$

so we update dict correspondingly

$\text{dict} = \{(-1: 0), (3, 1), (4, 2)\}$

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

Index

Verify sudoku board

we have grid 9×9 sudoku board. verify current state of board adhere to the rule of board.

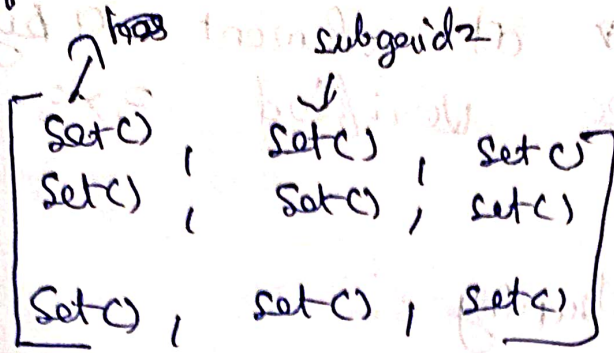
- 1) each row & column should be unique
- 2) each subgrid must be unique.

	0	1	2	3	4	5	6	7	8
0	3		6		5	8	4		
1	5	2						3	1
2		8	7						
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 (i.e.) i can use hashtable 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 3×3 is complex
- 5) To do that we are going to maintain 2D hashtable

for subgrid 1



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

7) in a row wise 0-2 belongs to subgrid-1
3-5 belong subgrid-2
6-8 belong subgrid-3

in column wise

0-2 → subgrid-1, subgrid-2, subgrid-3

3-5 → subgrid-4, 5, 6

6-8 → subgrid-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

row = 9

column = 9

$$[r/3][c/3] \Rightarrow [9/3][9/3] = [3][3]$$

which means ~~at~~ at element q by q
need to be verified 3×3 hashset

Zero Stepping

Given $m \times n$ matrix. if an element is zero set
it entire row & column as zero

This problem can be solved using hashset

STEP by STEP approach

- 1) Create two hashset for tracking the
column that need to be zero & row also
- 2) ~~row~~ row hashset to track column
column hashset to track row
- 3) loop the matrix one by one. if element
is zero then add row number to
row hashset and column number to
column hashset.
- 4) Then loop the matrix again. now
check ~~row~~ current row number present in
hashset or column number present in hashset.
- 5) if yes then make the current
element as zero

	0	1	2	3	4
0	2	3	8	6	7
1	1	2	3	0	4
2	5	0	6	7	8
3	9	10	11	12	13

now we have two hashset

$$\text{rowHashset} = \{ \}$$

$$\text{colHashset} = \{ \}$$

after loop the ~~the~~ matrix we will have like this

$$\text{rowHashset} = \{ 1, 3 \}$$

$$\text{colHashset} = \{ 1, 2 \}$$

Then loop the matrix again make the elements zero accordingly