13-06-2024

Test-I

Remove element

1) Aim:- An integer array nums and an integer val, remove all occurence of val in nums in place

Algorithm:

1) Take an function and with nums and vals as arguments
2) Put the constant number as '0' an
3) if nums != val then k=num and k is increment as 1. (k+=1)
4) Then return the k value.

I/P:- [3,2,2,3], val=3
O/P:- [2,2]

Time complexity = O(n-1)

2) Sudoku board:-

Aim:- Determine if a 9x9 Sudoku board is valid or not.

Algorithm:-

1) Take a function as sudoku with the arguments
2) using for loop take the rows and columns and its range as 9.
3) if board[i][j]=='.' the current number sudoku board is equal to board[i][j].
4) come out of the loop then if (9x9 boxes seen or (3x3 sub-boxes are seen). the return false
5) else return True.

sudoku solver:-

Aim:- solve a sudoku puzzle by filling the empty cells

Algorithm:-

1) Take a function with arguments (like num, val)

2) Take a loop for i in range(9). then

if board num[i][j] == board num[c]. then return false else True

3) Take other loop consideration of rows & columns

4) then take a if loop board[i][j] = " ".

5) put num as '123456789'.

then - if valid (nums, val):

board[i][j] = num.

if solve():

return True

else

return false.

I/P:-

O/P:-

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

count and say:-

Aim:- To determine of saying and conversion for digit string.

Algorithm:-

1) Take a function with arguments (like count)

2) Take a loop a "n==1" then return the

- Take the length of prev number

nd count and say(n-1).

Take of loop and count += 1 else

count = 1

of Prev(len) = Prev(len(n+1))

count = 1

else:

count = 1

Return prev length

I/P: n=4, O/P: "1211"

combination sum:

aim: An array of distinct integer candidates and target one integer then the sum of combination that ↓ integer

algorithm:

1) suppose the array list as some candidate
2) select the element that of from the arr. list.
3) Then the sum of combinations is equal to target number.
4) But same number can repeat more times
5) And then target element came separately from the list.

I/P: [2,3,6,7], target = 7

O/P: [2,2,3], [7]

combinations sum II :-

Aim:- determine The sol'n set must not contain duplicate combinations.

Algorithm:-

1) Take a array list of candidates and given the target as an integer.

2) Then the combinations of array list must equal to sum of integer target.

3) Like 1,1,6 in this way the combination will be formed.

4) Then find the how many combinations same from the list.

I/P:- [10,1,2,7,6,1,5], target = 8 o/p:- [1,1,6], [1,2,5], [1,7], [2,5]

Permutations II :-

Aim:- Determine how many permutations occur from a list.

Algorithm:-

1) Take an array of numbers as 'n'.

2) Then take the list how many times we swap the numbers from the list.

3) Suppose num = [1,2,3] then n1 = 3 because the length of array is = 3.

4) Then the make the array it can generate the Permutations.

5) And then result the list

I/P: [1,1,2], O/P: [[1,1,2],[1,2,1],[2,1,1]]

6) Maximum subarray:-

Aim:- The subarray which has the largest sum and return its sum

Algorithm:-

1) Suppose take a list of numbers.
2) Then divide the list of array in different subarray.
3) Take a list with four numbers each and find the largest sum among them.
4) Return the list of array.

I/P:- nums = [-2,1,-3,4,-1,2,1,5,4]

O/P:- 6