1]<https://practice.geeksforgeeks.org/problems/count-the-triplets4615/0>

class Solution:

def countTriplet(self, arr, n):

count=0

s=set(arr)

for i in range(n):

for j in range(i+1,n):

if arr[i]+arr[j] in s:

count+=1

return count

2] <https://practice.geeksforgeeks.org/problems/subarray-with-0-sum-1587115621/0>

class Solution{

static boolean findsum(int arr[],int n)

{

int preSum = 0;

HashSet<Integer> set = new HashSet<Integer>();

set.add(0);

for(int i=0;i<n;i++){

preSum += arr[i];

if(set.contains(preSum)) return true;

else set.add(preSum);

}

return false;

}

}

3] <https://practice.geeksforgeeks.org/problems/find-duplicates-in-an-array/0>

from collections import Counter

class Solution:

def duplicates(self, arr, n):

# code here

d={}

li=[]

if len(arr)==len(set(arr)):

li.append(-1)

d=Counter(arr)

for key,val in d.items():

if val>1:

li.append(key)

return sorted(li)

4] https://practice.geeksforgeeks.org/problems/minimum-swaps/0

class Solution:

def minSwaps(self, nums):

numIndx = [[nums[i],i] for i in range(len(nums))]

numIndx.sort()

i=0

count = 0

while i<len(nums):

j = numIndx[i][1]

if j==i:

i+=1

continue

numIndx[i], numIndx[j] = numIndx[j], numIndx[i]

count+=1

return count

5] <https://practice.geeksforgeeks.org/problems/permutations-of-a-given-string2041/0>

from itertools import permutations

class Solution:

def find\_permutation(self, S):

s=permutations(S)

li=["".join(i)for i in s]

lis=list(set(li))

lis.sort()