349. Intersection of Two Arrays

Intersection of Two Arrays

```
class Solution:
    def intersection(self, nums1: List[int], nums2: List[int]) -> List[int]:
        i=0
        j=0

        res=set()
        nums1.sort()
        nums2.sort()

        while(i<len(nums1) and j<len(nums2)):
            if nums1[i]<nums2[j]:
                i+=1
                continue

            if nums1[i]>nums2[j]:
                j+=1
                continue

            res.add(nums2[j])
            i+=1
            j+=1
        return res
```

922. Sort Array By Parity II

Sort Array By Parity II

```
class Solution:
    def sortArrayByParityII(self, A: List[int]) -> List[int]:
        A.sort()
        res=[]
        even=[]
        odd=[]
        for i in range(len(A)):
            if(A[i]%2==0):
                even.append(A[i])

            else:
                odd.append(A[i])

        for i in range(len(A)//2):
            res.append(even[i])
            res.append(odd[i])
        return res
```

976. Largest Perimeter Triangle

Largest Perimeter Triangle

```
class Solution:
    def largestPerimeter(self, A: List[int]) -> int:
        A.sort(reverse=True)

        for i in range(len(A)-2):
            if(A[i]<A[i+1]+A[i+2]):
                return A[i]+A[i+1]+A[i+2]
        return 0
```

1122. Relative Sort Array

Relative Sort Array

```
class Solution:
    def relativeSortArray(self, arr1: List[int], arr2: List[int]) -> List[int]:

        r = []  # initialize our return array

        for i in arr2:           # for each number in arr2
            while i in arr1:     # while that number is in arr1, we:
                r.append(i)      # add the number in question to our return
                arr1.remove(i)   # remove the number in question from arr1

        r.extend(sorted(arr1))   # when we're done, all that remains in arr1 are the numbers
                    # that weren't contained in arr2 so we sort those and add them to return
list
        return r
```

1370. Increasing Decreasing String

Increasing Decreasing String

```
class Solution:
    def sortString(self, s: str) -> str:
        s=list(s)
        result=''

        while(s):
            for letter in sorted(set(s)):
                s.remove(letter)
                result+=letter

            for letter in sorted(set(s),reverse=True):
                s.remove(letter)
                result+=letter

        return result
```

1491. Average Salary Excluding the Minimum and Maximum Salary

Average Salary Excluding the Minimum and Maximum Salary

```
class Solution:
    def average(self, salary: List[int]) -> float:
        max=salary[0]
        min=salary[0]

        avg=0

        for i in salary:
            if i>max:
                max=i

            elif i<min:
                min=i
            avg+=i

        return ((avg-max-min)/(len(salary)-2))
```

1502. Can Make Arithmetic Progression From Sequence

Can Make Arithmetic Progression From Sequence `class Solution: def canMakeArithmeticProgression(self, arr: List[int]) -> bool: arr.sort() diff=arr[1]-arr[0]

```
    for i in range(1,len(arr)):

        if(arr[i]-arr[i-1]==diff):
            continue

        else:
            return False

    return True

    `
```

1528. Shuffle String

Shuffle String

```
class Solution:
    def restoreString(self, s: str, indices: List[int]) -> str:
        res=list(s)
        l=len(s)

        for i in range(l):
            res[indices[i]]=s[i]
        return "".join(res)
```