Day 33

Insertion sort is a simple sorting algorithm that works the way we sort playing cards in our hand.

Algorithm:

```
insertion sort(arr,n)
    loop from i=1 to n-1
        Pick element arr[i] and insert it
        into sorted sequance arr[0,...i-1]
```

Implementation:

```
def insertionsortrecursive(a,n):
    if n<=0:
        return n
    insertionsortrecursive(a,n-1)
    last = a[n-1]
    j = n-2
    while (j<=0 and a[j]>last):
        a[j+1] = a[j]
        j -=1
    a[j+1] = last
```

Input :

```
a = [12, 11, 13, 5, 6]
n = len(a)
insertionSortRecursive(arr, n)
print(a)
```

Output :

[5, 6, 11, 12, 13]

Merge Sort

Merge sort is a divide and conquer algorithm. it divides the input array into two halves, call itself for the two halves and then merges two sorted halves.

Implementation :

```
def merge_sort(a):
    if len(a) > 1:
        mid = len(a)//2
        lefthalf = a[:mid]
        righthalf = a[mid:]
```

```
merge_sort(lefthalf)
merge_sort(righthalf)
i=0
k=0
j=0

while i< len(lefthalf) and j< len(righthalf):
    if lefthalf[i] < righthalf[j]:
        a[k] = lefthalf[i]
        i+=1
    else:
        a[k] = righthalf[i]
        j+=1
    k+=1

while i < len(lefthalf):
    a[k] = lefthalf[i]
    i +=1
    k +=1

while j < len(righthalf):
    a[k] = righthalf[j]
    j +=1
    k +=1
```

Input :

```
arr = [12, 11, 13, 5, 6, 7]

merge_sort(arr)

print (arr)
```

Output :

```
[5, 6, 7, 11, 12, 13]
```