**BINARY SEARCH:**

def binary\_search(arr, key):

start = 0

end = len(arr) - 1

while start <= end:

mid = (start + end) // 2

if arr[mid] == key:

return mid

elif key > arr[mid]:

start = mid + 1

else:

end = mid - 1

return -1

arr = [4, 5, 6, 7, 8, 9, 10, 11, 12]

key = 7

print(binary\_search(arr, key))

**MERGE SORT:**

def merge\_sort(arr, start=0, end=None):

if end is None:

end = len(arr) - 1

if start < end:

mid = (start + end) // 2

merge\_sort(arr, start, mid)

merge\_sort(arr, mid + 1, end)

merge(arr, start, mid, end)

def merge(arr, start, mid, end):

left = arr[start:mid + 1]

right = arr[mid + 1:end + 1]

i = j = 0

k = start

while i < len(left) and j < len(right):

if left[i] < right[j]:

arr[k] = left[i]

i += 1

else:

arr[k] = right[j]

j += 1

k += 1

while i < len(left):

arr[k] = left[i]

i += 1

k += 1

while j < len(right):

arr[k] = right[j]

j += 1

k += 1

arr = [5, 2, 4, 1, 3]

merge\_sort(arr)

print("Sorted array:", arr)

**QUICK SORT:**

def quick\_sort(arr):

if len(arr) <= 1:

return arr

left=[]

right=[]

equal=[]

pvt=arr[-1]

for i in arr:

if i < pvt:

left.append(i)

elif i > pvt:

right.append(i)

else:

equal.append(i) # i == pvt

print("povit:", pvt)

print("left sub array:", left)

print("right sub array:", right)

print("equal sub array:", equal)

return quick\_sort(left) + equal + quick\_sort(right)

arr = [23, 63, 44, 57, 12, 45, 36]

print(quick\_sort(arr))

**INSERTION SORT:**

def insertion\_sort(arr):

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

key = arr[i]

j = i - 1

while(j >= 0 and key < arr[j]):

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

return arr

arr = [30, 12, 4, 49, 45]

insertion\_sort(arr)

print(arr)

BOBBLE SORT:

def bobble\_sort(arr):

n=len(arr)

for i in range(n):

for j in range(0, n-i-1):

if arr[j] > arr[j+1]:

arr[j], arr[j + 1] = arr[j + 1], arr[j]

return arr

arr = [45, 8, 9, 3]

bobble\_sort(arr)

print(arr)