1.Binary search

def binary\_search(arr, target, low=0, high=None):

if high is None:

high = len(arr) - 1

if low > high:

return -1

mid = (low + high) // 2

if arr[mid] == target:

return mid

elif arr[mid] < target:

return binary\_search(arr, target, mid + 1, high)

else:

return binary\_search(arr, target, low, mid - 1)

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

target = 3

print(binary\_search(arr, target))

2.Merge sort

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = merge\_sort(arr[:mid])

right = merge\_sort(arr[mid:])

return merge(left, right)

def merge(left, right):

sorted\_arr = []

while left and right:

sorted\_arr.append(left.pop(0) if left[0] < right[0] else right.pop(0))

return sorted\_arr + left + right

arr = [38, 27, 43, 3, 9, 82, 10]

sorted\_arr = merge\_sort(arr)

print(sorted\_arr)

3.Quick sort

def quick\_sort(arr):

if len(arr) <= 1:

return arr

pivot = arr[len(arr) // 2]

return quick\_sort([x for x in arr if x < pivot]) + [pivot] + quick\_sort([x for x in arr if x > pivot])

arr = [3, 6, 8, 10, 1, 2, 1]

sorted\_arr = quick\_sort(arr)

print(sorted\_arr)