def binary\_search\_algo(arr ,target):

n=len(arr)

left=0

right=n-1

mid=(left+right)//2

while(left<right):

if arr[mid]==target:

return mid

if arr[mid]<target:

left=mid+1

if arr[mid]>target:

right=mid-1

return mid

arr=[2,3,4,5,6,7]

print(binary\_search\_algo(arr, 4))

def find\_max\_sum(nums, k):

cur\_sum=sum(nums[:k])

max\_sum=cur\_sum

for i in range(len(nums)):

cur\_sum=cur\_sum+nums[i]-nums[i-k]

max\_sum=max(cur\_sum, max\_sum)

return max\_sum/k

def search\_in\_the\_rotated\_array(nums ,target):

left=0#at index 0

n=len(nums) #length of the array

right=n-1 #last index

while(left<right):

mid=(left+right)//2 #calculate middle

if arr[mid]==target: #check whether the array is euqal to middle

return mid #return middle

#consider the left is sorted

if nums[left]<= arr[mid]:

if target>=nums[left] and target< arr[mid]:

right=mid-1 #now here we consider the left part contain the taget

else:

left=mid+1

else:

#consider the right array is sorted

if target> nums[mid] and target<=arr[right]:

left=mid+1 #now here we considerthe right part is sorted

else:

right=mid-1

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

print(search\_in\_the\_rotated\_array(arr,1))

insertion\_sort

def insertion\_sort(arr):

for i in range(len(array)):

j=i

while(j>0 and arr[j]<arr[j-1]):

arr[j-1] ,arr[j]=arr[j],arr[j-1]

j-=1

bubble sort

def bubble\_sort(arr):

for i in range(len(arr)):

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

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

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

return arr

def selection\_sort(arr):

for i in range(len(arr)):

min\_index=i

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

if arr[min\_index]>arr[j]:

min\_index=j

arr[min\_index],arr[i]=arr[i],arr[min\_index]

return arr

print(selection\_sort([23,4,5,6,6]))