**Answer1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Search Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| Bubble Sort | Ω(n) | θ(n^2) | O(n^2) |
| Selection Sort | Ω(n^2) | θ(n^2) | O(n^2) |
| Insertion Sort | Ω(n) | θ(n^2) | O(n^2) |
| Quick Sort | Ω(n log(n)) | θ(n log(n)) | O(n^2) |
| Merge Sort | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) |

**Answer2**

#Selection Sort

arr= [3,5,2,6,8,4,1,9,7]

L= len(arr)

for i in range(0,L):

pos=i

mini=arr[i]

for j in range(i,L):

if(arr[j]<mini):

mini=arr[j]

pos=j

temp=arr[i]

arr[i]=arr[pos]

arr[pos]=temp

print(arr)

**Answer3**

#pop using stack - array implementation - size(4)

#custom made scenario

#array elements and top definition starts

arr=[1,2,3,4]

top= 3

#definition ends here

def pop():

global arr

global top

if(top<0):

print("Stack UnderFlow")

else:

print("value popped= "+str(arr[top]))

top=top-1

#user control starts here

pop()

**Answer4**

#dequeue implementation using array

arr=[1,2,3,4,5,6,7,8,9]

#array with 9 elements

#start and end flags are given different locations

startLimit = 0

endLimit =len(arr)

start=2

end=5

def printDequeue():

global start,end

output=""

for i in range(start,end+1):

output= output+ str(arr[i])+"\t"

print(output)

def addAtStart(x):

global start,startLimit

if((start-1)<startLimit):

print("Dequeue full at start")

else:

start=start-1

arr[start]=x

print("Value: "+str(x)+" added at pos: "+str(start)+"(start)")

def addAtEnd(x):

global end,endLimit

if((end+1)>=endLimit):

print("Dequeue full at end")

else:

end=end+1

arr[end]=x

print("Value: "+str(x)+" added at pos: "+str(end)+"(end)")

def removeFromStart():

global start,end

if((start+1)>end):

print("Dequeue is empty")

else:

val=arr[start]

start=start+1

print("Value: "+str(val)+" removed from start")

def removeFromEnd():

global start,end

if((end-1)<start):

print("Dequeue is empty")

else:

val=arr[end]

end=end-1

print("Value: "+str(val)+" removed from end")

#corner test cases not considered in this example

#iems count flag not used

#user control starts here

printDequeue()

addAtStart(0)

printDequeue()

addAtEnd(7)

printDequeue()

removeFromStart()

printDequeue()

removeFromEnd()

printDequeue()