Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 6

Problem : Create an application for Ganpat University to list and sort student's marks. Repeat the experiment for different values of n. The number of elements in the list to be sorted and plot a graph of the comparison count of any 3 sorting algorithms.

Code :

```
import random
import matplotlib.pyplot as plt
from matplotlib import use
use('Qt5Agg')
import YSL_io as YSL


# Merge Sort
def merge_sort(ysl):
def merge(arr, l, m, r, count):
n1 = m - l + 1
n2 = r - m


L = [0] * n1
R = [0] * n2
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
for i in range(n1):
L[i] = arr[l + i]
for j in range(n2):
R[j] = arr[m + 1 + j]

i = 0
j = 0
k = l

while i < n1 and j < n2:
count[0] += 1
if L[i] ⩽ R[j]:
arr[k] = L[i]
i += 1
else:
arr[k] = R[j]
j += 1
k += 1

while i < n1:
arr[k] = L[i]
i += 1
k += 1
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
while j < n2:
arr[k] = R[j]
j += 1
k += 1

def sort(arr, l, r, count):
if l < r:
m = l + (r - l) // 2
sort(arr, l, m, count)
sort(arr, m + 1, r, count)
merge(arr, l, m, r, count)

count = [0]
sort(ysl, 0, len(ysl) - 1, count)
return count[0]

# Quicksort
def quick_sort(ysl):
def partition(arr, low, high, count):
i = low - 1
pivot = arr[high]

for j in range(low, high):
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
count[0] += 1
if arr[j] ⩽ pivot:
i += 1
arr[i], arr[j] = arr[j], arr[i]

arr[i + 1], arr[high] = arr[high], arr[i + 1]
return i + 1

def sort(arr, low, high, count):
if low < high:
pi = partition(arr, low, high, count)
sort(arr, low, pi - 1, count)
sort(arr, pi + 1, high, count)

count = [0]
sort(ysl, 0, len(ysl) - 1, count)
return count[0]

# Heapsort
def heap_sort(ysl):
def heapify(arr, n, i, count):
largest = i
l = 2 * i + 1
r = 2 * i + 2
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
if l < n and arr[l] > arr[largest]:
largest = l
if r < n and arr[r] > arr[largest]:
largest = r

if largest ≠ i:
count[0] += 1
arr[i], arr[largest] = arr[largest], arr[i]
heapify(arr, n, largest, count)

def sort(arr, count):
n = len(arr)
for i in range(n // 2 - 1, -1, -1):
heapify(arr, n, i, count)
for i in range(n - 1, 0, -1):
arr[i], arr[0] = arr[0], arr[i]
heapify(arr, i, 0, count)

count = [0]
sort(ysl, count)
return count[0]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
# Function to plot the comparison graph for a data category
def compare(size, algos, labels, colors, data_category="random"):
plt.figure()
plt.xlabel("Data Size")
plt.ylabel("Count of Iterations")
plt.title(f"\nSorting Algorithm Comparison for {data_category} data\n",
fontsize=15)

for i, algo in enumerate(algos):
iterations = []
for s in size:
data = [random.randint(1, 100) for y in range(s)]
if data_category == "ascending":
data.sort()
elif data_category == "descending":
data.sort(reverse=True)
iterations.append(algo(data))
# print(f'\n{data}')

plt.plot(size, iterations, marker="o", label=labels[i], color=colors[i],
linewidth=2.5)
if i==0:
print('\nFor', end=' ')
YSL.printMGNTA(data_category.capitalize(), end=' ')
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
print('data, number of iterations using', end=' ')
YSL.printMGNTA(labels[i],end=' ')
print('for sizes -', end=' ')
YSL.printMGNTA(size, end=' ')
print(' :', end=' ')
YSL.printMGNTA(iterations)
elif i==1:
print('For', end=' ')
YSL.printRED(data_category.capitalize(), end=' ')
print('data, number of iterations using', end=' ')
YSL.printRED(labels[i],end=' ')
print('for sizes -', end=' ')
YSL.printRED(size, end=' ')
print(' :', end=' ')
YSL.printRED(iterations)
else:
print('For', end=' ')
YSL.printBLU(data_category.capitalize(), end=' ')
print('data, number of iterations using', end=' ')
YSL.printBLU(labels[i],end=' ')
print('for sizes -', end=' ')
YSL.printBLU(size, end=' ')
print(' :', end=' ')
YSL.printBLU(iterations)
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

```
plt.legend()
plt.grid(True)



size = [10, 20, 30, 50, 80]

algos = [merge_sort, quick_sort, heap_sort]
methods = ["Merge Sort", "Quick Sort", "Heap Sort"]

colors = ["#8839ef", "#dd7878", "#179299"]

compare(size, algos, methods, colors, "ascending")
compare(size, algos, methods, colors, "descending")
compare(size, algos, methods, colors)



plt.show()
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
AAD Practical 6

Output :