**Assignment NO 3**

**Name:Najeebullah**

**Sap id:45824**

import time

import random

def measure\_time(func, arr):

"""Measure execution time of a function"""

start\_time = time.time()

func(arr.copy()) # Pass a copy to preserve original array

end\_time = time.time()

return (end\_time - start\_time) \* 1000 # Convert to milliseconds

def bubble\_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

def selection\_sort(arr):

n = len(arr)

for i in range(n):

min\_idx = i

for j in range(i+1, n):

if arr[j] < arr[min\_idx]:

min\_idx = j

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

return arr

def merge\_sort(arr):

if len(arr) <= 1:

return arr

mid = len(arr) // 2

left = arr[:mid]

right = arr[mid:]

merge\_sort(left)

merge\_sort(right)

i = j = k = 0

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

return arr

def quick\_sort(arr):

if len(arr) <= 1:

return arr

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

left = [x for x in arr if x < pivot]

middle = [x for x in arr if x == pivot]

right = [x for x in arr if x > pivot]

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

# Generate test arrays

size = 1000

# Best case scenario (already sorted)

best\_case = list(range(size))

# Average case scenario (random numbers)

average\_case = random.sample(range(size), size)

# Worst case scenario (reverse sorted)

worst\_case = list(range(size, 0, -1))

# Test arrays

test\_arrays = {

"Best Case": best\_case,

"Average Case": average\_case,

"Worst Case": worst\_case

}

# Sorting algorithms

algorithms = {

"Bubble Sort": bubble\_sort,

"Selection Sort": selection\_sort,

"Merge Sort": merge\_sort,

"Quick Sort": quick\_sort

}

# Run performance tests

results = {}

for algo\_name, algo\_func in algorithms.items():

results[algo\_name] = {}

for case\_name, arr in test\_arrays.items():

execution\_time = measure\_time(algo\_func, arr)

results[algo\_name][case\_name] = execution\_time

# Print results

print("\nPerformance Results (in milliseconds):")

print("-" \* 60)

print(f"{'Algorithm':<15} {'Best Case':>12} {'Average Case':>15} {'Worst Case':>12}")

print("-" \* 60)

for algo\_name in algorithms:

print(f"{algo\_name:<15} {results[algo\_name]['Best Case']:>12.4f} "

f"{results[algo\_name]['Average Case']:>15.4f} "

f"{results[algo\_name]['Worst Case']:>12.4f}")

print("-" \* 60)