NAME: RIMSHA LARAIB
SEAT NO.: B21110006107
CLASS: BSCS (4TH YEAR)

Assignment # 2

Question: Generate 500 unique random numbers than apply any two searching and sorting algorithms on them and compare for efficiency.

Searching

Code:

```
import random
import time

# Step 1: Generate 499 unique random numbers
data = random.sample(range(1, 10000), 499)

print("\nGenerated Numbers (first 5 and last 5 shown):")
print(data[:5], "...", data[-5:])
target = int(input("\nEnter a number to search (between 1 and 9999): "))
def insertion_sort(arr):
    a = arr.copy()
    for i in range(1, len(a)):
        key = a[i]
        j = i - 1
        while j >= 0 and a[j] > key:
            a[j + 1] = a[j]
            j -= 1
        a[j + 1] = key
    return a
def linear_search(arr, target):
    for i in range(len(arr)):
        if arr[i] == target:
            return i
    return -1
def binary_search(arr, target):
    low, high = 0, len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1
```

```
sorted_insertion = insertion_sort(data)
start_linear = time.perf_counter()
linear_index = linear_search(data, target)
end_linear = time.perf_counter()
linear_time = end_linear - start_linear
start_binary = time.perf_counter()
binary_index = binary_search(sorted_insertion, target)
end_binary = time.perf_counter()
binary_time = end_binary - start_binary
print(f"Target Number: {target}")
print("\n Searching Algorithm Comparison:")
print(f"Linear Search: Index = {linear_index}, Time = {linear_time:.10f}
seconds")
print(f"Binary Search: Index = {binary_index}, Time = {binary_time:.10f}
seconds")
if linear_index == -1:
    print("Number not found using Linear Search.")
else:
    print("Number found using Linear Search.")
if binary_index == -1:
    print("Number not found using Binary Search.")
else:
    print("Number found using Binary Search.")
print("\n Efficiency Comparison:")
if linear_time < binary_time:
    diff = binary_time - linear_time
    print(f"Linear Search is faster by {diff:.10f} seconds.")
elif binary_time < linear_time:
    diff = linear_time - binary_time
    print(f"Binary Search is faster by {diff:.10f} seconds.")
else:
    print("Both searches took the same time.")
```

Output:

```
Generated Numbers (first 5 and last 5 shown):
[6186, 5527, 3158, 2475, 3681] ... [7655, 6915, 5671, 7088, 4637]

Enter a number to search (between 1 and 9999): 7655
Target Number: 7655

 Searching Algorithm Comparison:
Linear Search: Index = 494, Time = 0.0000216000 seconds
Binary Search: Index = 386, Time = 0.0000085000 seconds
Number found using Linear Search.
Number found using Binary Search.

 Efficiency Comparison:
Binary Search is faster by 0.0000131000 seconds.
```

Sorting

Code:

```
import random
import time

# Generate 500 unique random numbers
data = random.sample(range(1, 10000), 500)

def bubble_sort(arr):
    a = arr.copy()
    n = len(a)
    for i in range(n):
        for j in range(0, n-i-1):
            if a[j] > a[j+1]:
                a[j], a[j+1] = a[j+1], a[j]
    return a

def insertion_sort(arr):
    a = arr.copy()
    for i in range(1, len(a)):
        key = a[i]
        j = i - 1
        while j >= 0 and a[j] > key:
            a[j + 1] = a[j]
            j -= 1
        a[j + 1] = key
    return a
start_bubble = time.perf_counter()
sorted_bubble = bubble_sort(data)
end_bubble = time.perf_counter()
bubble_time = end_bubble - start_bubble
start_insertion = time.perf_counter()
sorted_insertion = insertion_sort(data)
end_insertion = time.perf_counter()
insertion_time = end_insertion - start_insertion

print("\n Sorting Algorithm Comparison:")
print(f"Bubble Sort Time:    {bubble_time:.6f} seconds")
print(f"Insertion Sort Time: {insertion_time:.6f} seconds")

preview = sorted_bubble[:5] + ["..."] + sorted_bubble[-5:]
print("\n Bubble Sorted Output:")
print(preview)

preview2 = sorted_insertion[:5] + ["..."] + sorted_insertion[-5:]
print("\n Insertion Sorted Output:")
```

```
print(preview2)

print("\n Efficiency Comparison:")
if bubble_time < insertion_time:
    print(" Bubble Sort is faster than Insertion Sort.")
elif insertion_time < bubble_time:
    print(" Insertion Sort is faster than Bubble Sort.")
else:
    print(" Both sorting algorithms took the same time.")
```

Output:

```
 Sorting Algorithm Comparison:
Bubble Sort Time:    0.009275 seconds
Insertion Sort Time: 0.003245 seconds

 Bubble Sorted Output:
[17, 45, 78, 105, 115, '...', 9885, 9944, 9958, 9967, 9993]

 Insertion Sorted Output:
[17, 45, 78, 105, 115, '...', 9885, 9944, 9958, 9967, 9993]

 Efficiency Comparison:
 Insertion Sort is faster than Bubble Sort.
```