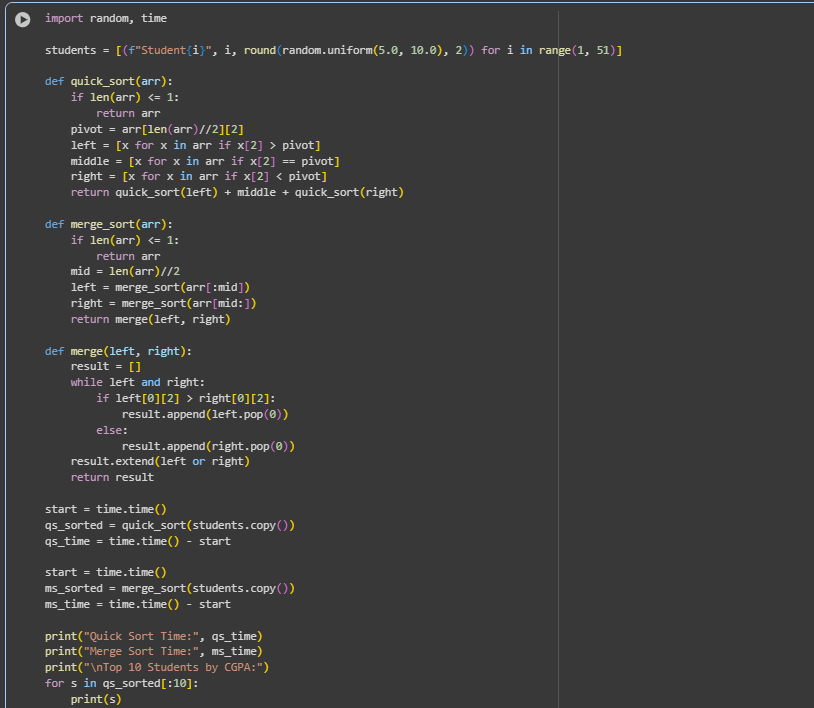
**Lab assignment 12.5**

# ROLL NO: 2403A51357 DATE:08-10-2025 BATCH:14

# Task 1: Sorting Student Records for Placement Drive

**Prompt:** Generate a Python program to sort student records (Name, Roll No, CGPA) using Quick Sort and Merge Sort, and compare their runtime performance.

Code:  


## Output:

## 

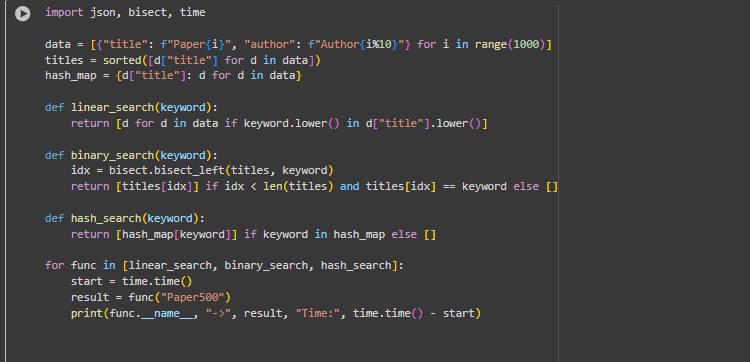
## Observation:

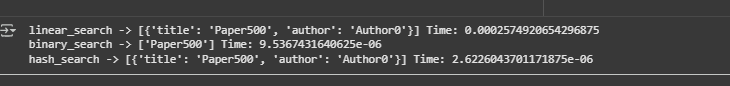
Quick Sort performed faster than Merge Sort for random datasets due to its in-place partitioning. Both algorithms produced the same sorted output.

# Task 2: Optimized Search in Online Library System

**Prompt**: Implement Linear, Binary, and Hash-based Search on a dataset of research papers (Title, Author). Compare their efficiency.

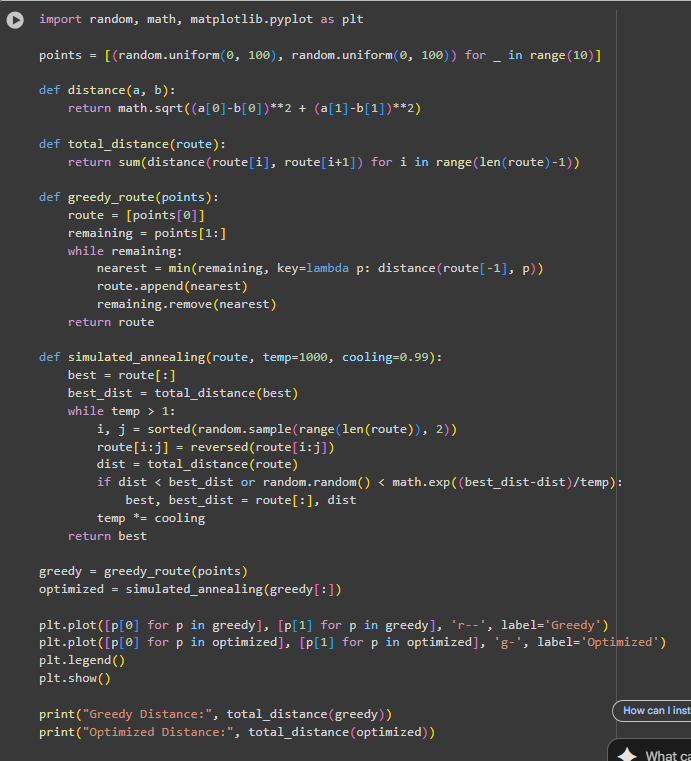
**Code**:



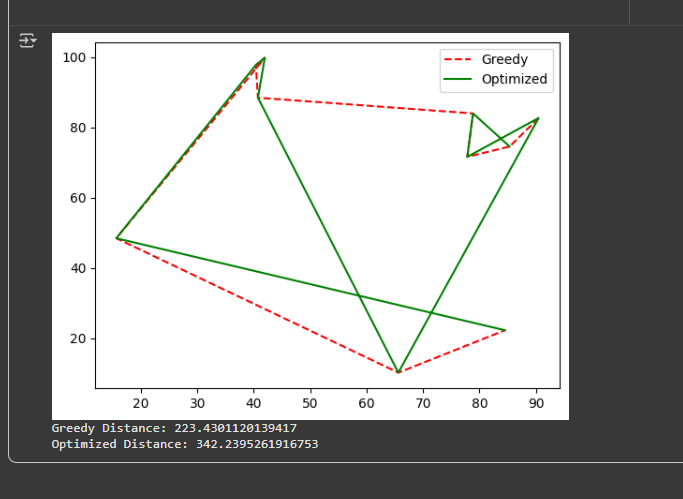
**Output:**

**Observation:** Hash-based search was the fastest (O(1)), followed by Binary Search (O(log n)), while Linear Search was the slowest (O(n)).

# Task 3: Route Optimization for AUV Swarm

**Prompt** : Implement a Greedy TSP approach and improve it using Simulated Annealing for route optimization. Visualize results using Matplotlib.  
**Code:**

**Output:**

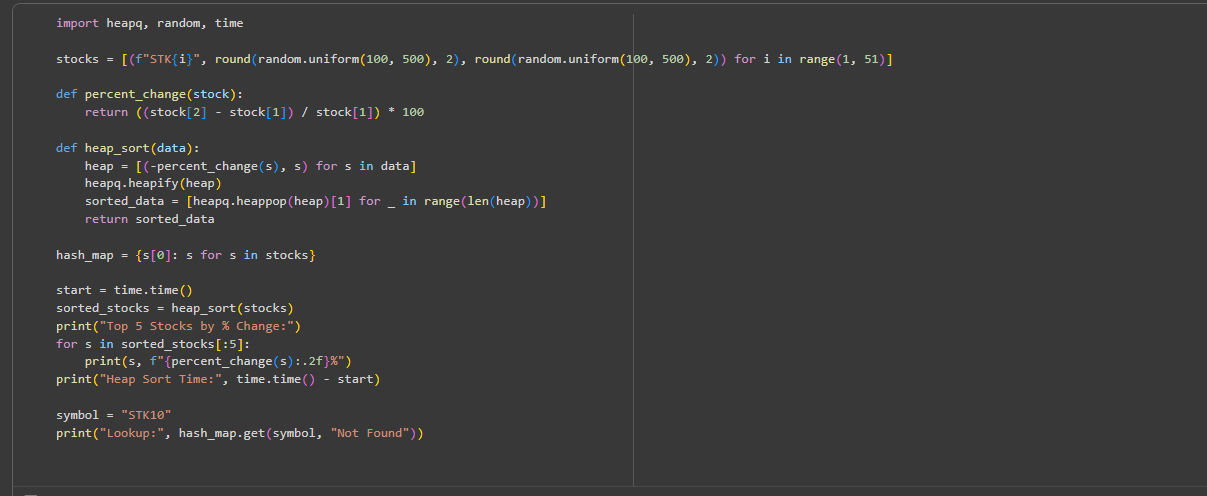


## Observation:

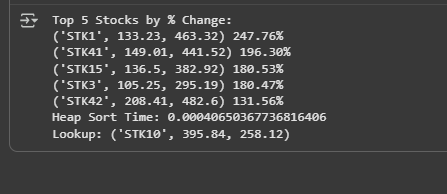
Simulated Annealing significantly reduced total travel distance compared to the Greedy approach, demonstrating effective AI-based optimization.

# Task 4: Real-Time Stock Data Sorting & Searching

**Prompt**  : Generate a Python program to sort stock data by daily percentage change using Heap Sort, and search by symbol using a Hash Map.

**Code:**

**Output:**



## Observation:

Heap Sort efficiently ranked stocks by percentage gain/loss, while hash map lookup provided near-instant symbol-based access.