NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

SC1007 Data Structures and Algorithms

Lab 5: Analysis of Algorithms

Q1 Given an array of n elements, try Bubble Sort and Merge Sort, and compare their execution times for different input sizes. You can generate an array of random numbers (e.g., size 1,000, 10,000, 100,000). Use the time module of Python to measure execution time. Plot the results of execution time for different input sizes.

Q2 Try the recursive and iterative Fibonacci. Measure execution time for different n values (e.g., 20, 30, 40, 45). Plot the results of execution time for different input. What is the time complexity for each algorithm?

Q3 Suppose you are a treasure hunter exploring an ancient cave filled with valuable artifacts. You have a backpack with a limited weight capacity W, and you must decide which artifacts to take to maximize your total value. Each artifact has a weight w_i (how heavy it is) and a value v_i (how much gold it's worth). However, you cannot take a fraction of an artifact, i.e., you must take either the whole artifact or leave it behind. Your goal is to maximize the total value of your loot while staying within the weight limit of your backpack.

Implement a brute force approach to try all possible subsets of artifacts. Measure execution time for different backpack capacities and artifacts values and weights shown in q3input.txt. Plot the results of execution time for different inputs and analyze how the execution time increases with the number of artifacts. For example, for the following input:

```
50
10 60
20 100
30 120
5 30
15 80
```

The first line indicates the backpack's capacity, which is 60. Each subsequent line represents an artifact's weight and value. For example, artifact 1 has a weight of 10 and a value of 60. In total, there are 5 artifacts.

The output will be:

```
Maximum Value: 270
Selected Items (weight, value): [(10, 60), (20, 100), (5, 30), (15, 80)]
Total Weight Used: 50
Execution Time: 0.00000 sec
```