**Assignment 4**

**DCIT 204**

**Joshua Danso- 11047613**

**Ewurafua Quansah - 11302223**

**Seth Osei Yaw Akoto - 11249336**

**Ekow Essel Haizel Davis- 11083926**

**Emmanuella Animwaa Kumah - 11090094**

**Caleb Oblitei Tetteh- 11117313**

**David Asamoa Berko - 11245046**

**1. Bubble Sort**

* **Time Complexity**:
  + Best Case: O(n) (when the array is already sorted)
  + Average and Worst Case: O(n^2)
* **Space Complexity**: O(1) (in-place sorting)

**2. Polynomial Evaluation (Naive Approach)**

* **Time Complexity**: O(n), where n is the degree of the polynomial
* **Space Complexity**: O(1) (only a few extra variables used)

**3. Brute Force String Matching**

* **Time Complexity**:
  + Best Case: O(m) (when the pattern is found at the beginning)
  + Average and Worst Case: O(m⋅n), where m is the length of the text and n is the length of the pattern
* **Space Complexity**: O(1) (no extra space required apart from variables)

**4. Subset Sum Problem (Recursive Approach)**

* **Time Complexity**: O(2^n), where n is the number of elements in the set
* **Space Complexity**: O(n) (due to recursion stack)

**5. Traveling Salesman Problem (Brute Force Approach)**

* **Time Complexity**: O(n!), where n is the number of cities
* **Space Complexity**: O(n) (due to recursion stack and visited array)

**6. Knapsack Problem (Dynamic Programming Approach)**

* **Time Complexity**: O(n⋅W), where n is the number of items and W is the capacity of the knapsack
* **Space Complexity**: O(n⋅W) (due to the DP table)

**7. Selection Sort**

* **Time Complexity**:
  + Best, Average, and Worst Case: O(n^2)
* **Space Complexity**: O(1) (in-place sorting)

Meeting Attendance for Tuesday, 18th June, 2024:

Joshua- Present

Caleb- Present

Seth- Present

Ewurafua- Present

Emmanuella- Present

Ekow- Present

David-Present