**Assignment 1 – Report**

**Data Structure & Algorithms**

**Dat Duong – 100886108**

**­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Introduction:**

* The online shopping platforms require sophisticated data structures and algorithms to efficiently manage large volume of product data.
* The objective of this assignment is to employ fundamental data structures for effective data management and to implement basic sorting algorithms for sorting product data.

1. **Snapshots of the Running Program:**

Menu Options:

**A screen shot of a computer

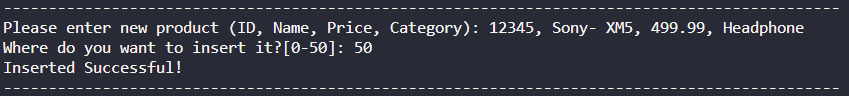
Description automatically generated**

* Load:

A screenshot of a computer

Description automatically generated

* Insert:



A screen shot of a computer

Description automatically generated

* Update:

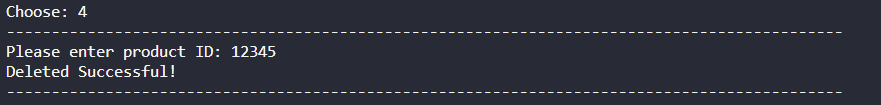
A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Delete:



A screenshot of a computer

Description automatically generated

* Search:

A screen shot of a computer

Description automatically generated

* Bubble Sort:

Average Case (original data):

Time taken: 0.00100136 seconds, Space used: 584 bytes.

Time Complexity: O(n^2)

A screen shot of a computer screen

Description automatically generated

Best Case (sorted):

Time taken: 0.00000000 seconds, Space used: 584 bytes.

Time Complexity: O(n)

A screen shot of a computer screen

Description automatically generated

Worst Case (reversed):

Time taken: 0.00107980 seconds, Space used: 584 bytes.

Time Complexity: O(n^2)

A screen shot of a computer screen

Description automatically generated

1. **Complexity Analysis Report:**

In the best-case scenario, when the data is already sorted, the time complexity of sorting algorithms can be as low as O(n) because the algorithm only need to make one pass through the data to confirm it's sorted.

In the average case, the time complexity is O(n^2) because the algorithm needs to make multiple passes through the data, comparing adjacent elements and swapping them if they are in the wrong order.

In the worst case, when the data is sorted in reverse order, the time complexity is also O(n^2) because the algorithm needs to make the maximum number of passes and swaps to correctly order the data.

1. **Code Implementation:**

* GitHub: <https://github.com/datduong1205/Product-Management>

1. **Conclusion:**

This assignment improves my understanding of data structures, sorting algorithms, and algorithm complexity analysis, which are essential for optimizing the performance of online shopping platforms.