**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Algorithms & Constructs |
| **Assessment Title:** | System Modelling & Build |
| **Lecturer Name:** | Mr. Muhammad Iqbal |
| **Student Full Name:** | Buyantugs Lkhagvadorj |
| **Student Number:** | 2024578 |
| **Assessment Due Date:** | Saturday 10th May 2025 - 23:59) |
| **Date of Submission:** | Saturday 10th May 2025 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Contents

[Problem Statement 2](#_Toc197711788)

[Algorithm Selection and Justification 3](#_Toc197711789)

[Solution Design and Features 3](#_Toc197711790)

[Strengths 4](#_Toc197711791)

[Weaknesses 4](#_Toc197711792)

[Alternatives Considered 4](#_Toc197711793)

[Conclusion 4](#_Toc197711794)

Report - Algorithms & Constructs

# Problem Statement

This project models a Police Station system with a goal to simulate real-world roles and department structures. The system must support reading applicant data from a file, generating random personnel, sorting and searching employee records, and allowing user interaction via a console menu.

The biggest challenges were:

- Designing a flexible structure for roles like Manager, Department, and Employee

- Ensuring smooth data input/output with file handling

- Choosing efficient algorithms for sorting and searching

- Maintaining a clean, modular codebase using best practices

# Algorithm Selection and Justification

Recursive Insertion Sort:

- Simple to implement recursively and ideal for small datasets (under 100 records).

- Works well for data read from Applicants\_Form.txt which is not large.

- Avoided complex algorithms like Merge Sort due to simplicity and space efficiency.

Binary Search (Exact Match):

- Fast O(log n) performance for sorted lists.

- Suitable for frequent, exact queries.

Partial Search (Linear):

- Enables user-friendly flexible search with substrings.

- Though linear (O(n)), acceptable for small datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Recursive? | Avg. Complexity | Space | Project Fit |
| Merge Sort | Yes | O(n log n) | O(n) | Overqualified |
| Quick Sort | Yes | O(n log n) | O(log n) | Risk of O(n²) worst case |
| Heap Sort | No | O(n log n) | O(1) | Doesn't meet recursion req |
| Insertion Sort | Yes | O(n²) | O(1) | Ideal fit |

# Solution Design and Features

- Employee, ManagerType, Department, and EmployeeRole enums and classes.

- Enum-based console menu for sorting, searching, adding, and generating data.

- Data read from Application\_Form.txt and assigned randomly to employee attributes.

- Search options support both exact and partial matches.

- Modular code organized within a CA\_2 package.

# Strengths

- Clear modular structure.

- Fulfills recursive and binary algorithm requirements.

- Easy-to-use console interface.

- Supports both real and generated data.

- Validates inputs to ensure correctness.

# Weaknesses

- Linear partial search is slower on large datasets.

- Recursive insertion sort unsuitable for large data.

- Data is not persisted to file or database after program exits.

# Alternatives Considered

- Merge Sort or Quick Sort: more complex, not needed for small data.

- Trie structure: too complex for this scope.

- HashMap for fast lookup: avoided to comply with recursive and binary search requirements.

# Conclusion

The Police Station system meets the design goals through appropriate algorithm selection and structured coding practices. It demonstrates recursion, binary search, and user interaction effectively, forming a solid base for future enhancements.