SIT221

DATA STRUCTURES AND ALGORITHMS

Learning Summary Report

JASVEENA
224001588

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment	✓	✓		

Minimum Pass Checklist

	Included
Learning Summary Report	✓
Pass tasks completed (submitted and discussed)	✓

Higher Grade Checklist (in addition to Pass Checklist)

	Included
Credit tasks completed (submitted and discussed)	✓
Distinction tasks completed (submitted and discussed)	
HD tasks completed (submitted and discussed)	

Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature: **JASVEENA**

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for SIT221 DATA STRUCTURES AND ALGORITHMS e to a **credit** level.

I believe I should receive this grade because I have consistently shown a good understanding and application of the core concepts taught in this unit. I completed all required pass-level tasks and extended myself by completing multiple credit-level tasks, including **2.3C (Advanced Complexity Questions)**, **4.3C (Rescuing Mario's Black Cat)**, and **6.2C (Binary Heap Implementation)**. These tasks demonstrate that I can go beyond minimal requirements and apply algorithmic thinking to solve problems effectively.

- For **ULO1**, I evaluated memory usage and time complexity across a wide range of solutions, and during performance analysis of different sorting and search algorithms.
- For **ULO2**, I implemented various data structures such as Vectors , Binary Search , Skip Lists , Doubly Linked Lists , Binary Heaps , and AVL Trees, using appropriate algorithms to solve each problem efficiently.
- For **ULO3**, I documented problem constraints, design trade-offs, and algorithm choices in the form of code comments and video explanations for each practical task.

Reflection

The most important things I learnt:

- The relationship between **algorithmic complexity** and real-world performance.
- How to **choose appropriate data structures** based on the requirements and constraints of a problem.
- Practical coding skills in **C#**, including the implementation of abstract data types and debugging complex logic.
- The importance of documenting **design decisions** and analyzing **space/time trade-offs**.

The things that helped me most were:

- The **step-by-step guided tasks**, which gradually increased in complexity.
- The **video demonstrations and explanations**, which clarified how to approach implementation challenges.
- Constant feedback and **Doubtfire task checklists**, which helped me track progress and fix issues early.

I found the following topics particularly challenging:

- Implementing the **Binary Heap** and ensuring correctness with **DownHeap** and **UpHeap** logic .
- Calculating and maintaining **balance factors** in **AVL Trees** and choosing the correct rotations .
- Understanding **Skip Lists**, especially probabilistic level generation and search logic

I found the following topics particularly interesting:

- The **AVL Tree balancing mechanisms** were fascinating in how they maintain order efficiently.
- **Graph traversal algorithms** in Task 8.1P were exciting to apply, especially in real-world analogies like routing or social networks.
- The design and debugging of **custom comparers** for sorting algorithms.

I feel I learnt these topics, concepts, and/or tools really well:

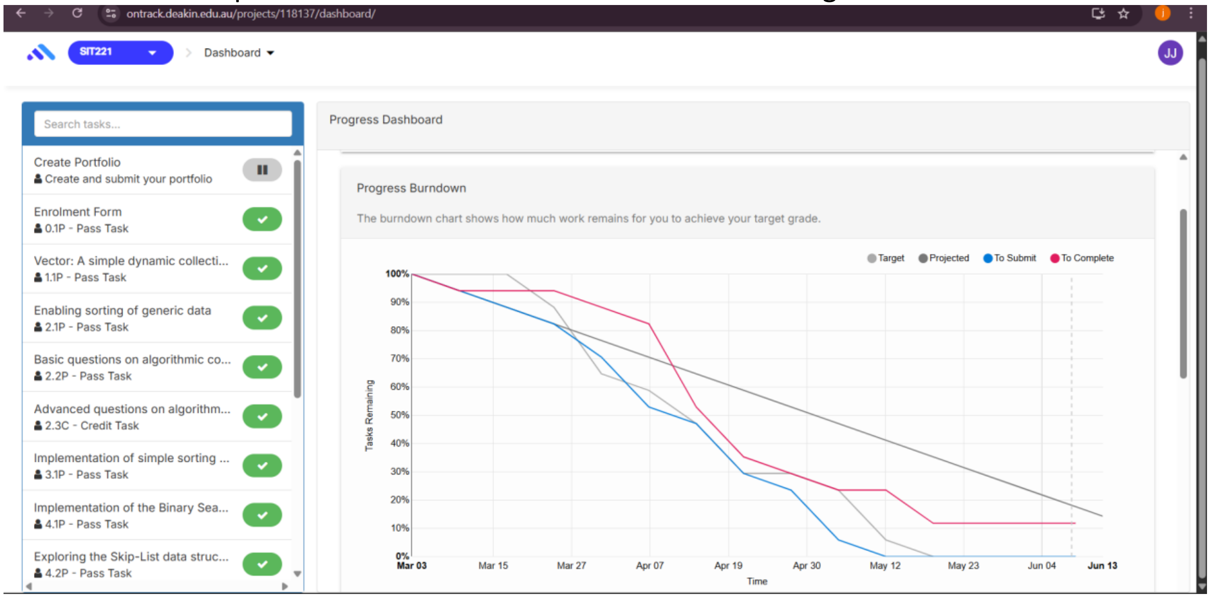
- **Sorting algorithms** and how to implement them from scratch .
- **Binary Search** logic and its conditions.
- Designing and building a **Vector<T>** generic class .
- Implementing a **Doubly Linked List** with full node manipulation and null safety .

I still need to work on the following areas:

- Getting more fluent with **advanced trees and rotations**, particularly in AVL trees.
- Enhancing my skills in **graph algorithms** beyond BFS and DFS.
- Writing **unit tests** for my data structures to catch edge cases faster.

My progress in this unit was ...:

From my perspective, I paced the unit steadily and increased my engagement during the mid and late stages. Early progress was slower, as I was adjusting to the technical complexity. However, once I gained confidence, especially around Task 3.1P and beyond, my progress became consistent, with timely submissions and attention to feedback. The graph reflects a clear improvement in commitment and understanding.


ontrack.deakin.edu.au/projects/118137/dashboard/

SIT221 > Dashboard

Search tasks...

Create Portfolio
Create and submit your portfolio

Enrolment Form
0.1P - Pass Task

Vector: A simple dynamic collecti...
1.1P - Pass Task

Enabling sorting of generic data
2.1P - Pass Task

Basic questions on algorithmic co...
2.2P - Pass Task

Advanced questions on algorithm...
2.3C - Credit Task

Implementation of simple sorting ...
3.1P - Pass Task

Implementation of the Binary Sea...
4.1P - Pass Task

Exploring the Skip-List data struc...
4.2P - Pass Task

Progress Dashboard

Progress Burndown

The burndown chart shows how much work remains for you to achieve your target grade.

This unit will help me in the future:

The skills I've developed will be vital in future units and in industry roles that require algorithm design and performance-aware coding. The ability to **analyze complexity**, implement custom **data structures**, and solve **realistic problems** has strengthened my core programming ability, which is essential for any role involving backend development, system design, or technical interviews.

If I did this unit again I would do the following things differently:

- Start the **credit tasks earlier**, giving more time to test and revise.
- Spend more time reviewing **sample solutions** and comparing alternative strategies.
- Collaborate more with peers to discuss solution strategies and get broader perspectives.

Other…:

SIT221 has been a foundational unit in helping me build both theoretical and practical skills in data structures and algorithms. I appreciated how each task built upon the previous, creating a cohesive learning journey from simple vectors to complex trees and heaps.