Study Point Assignment

By Daniel Ellebæk

Overall description

This assignment is 20 study points worth.

There are 5 assignments to choose from and 3 needs to be chosen.

Write 1 page of text (2400 characters without spaces) per assignment chosen to document your thoughts and research of the algorithm.

It is allowed to be two in a group.

Task 1

Exercise 1:

Title: Analyzing Time Complexity of a Linear Search Algorithm

Objective: To understand the concept of big O notation and to analyze the time complexity of a linear search algorithm.

Instructions:

- Research the concept of big O notation and its importance in analyzing the time complexity of algorithms.
- Implement a linear search algorithm in a programming language of your choice.
- Analyze the time complexity of your implementation using big O notation.

Task 2

Exercise 2:

Title: Analyzing Time Complexity of a Bubble sort Algorithm

Objective: To understand the concept of big O notation and to analyze the time complexity of a bubble sort algorithm.

Instructions:

- Research the concept of big O notation and its importance in analyzing the time complexity of algorithms.
- Implement a bubble sort algorithm in a programming language of your choice (e.g. C++, Python, Java).
- Analyze the time complexity of your implementation using big O notation.

Task 3

Exercise 3:

Title: Analyzing Time Complexity of a Stack

Objective: To understand the concept of big O notation and to analyze the time complexity of a stack data structure.

Instructions:

- Research the concept of big O notation and its importance in analyzing the time complexity of data structures.
- Implement a stack data structure in a programming language of your choice (e.g. C++, Python, Java).
- Analyze the time complexity of the push, pop and peek operations using big O notation.

Task 4

Exercise 4:

Title: Analyzing Time Complexity of a Queue

Objective: To understand the concept of big O notation and to analyze the time complexity of a queue data structure.

Instructions:

- Research the concept of big O notation and its importance in analyzing the time complexity of data structures.
- Implement a queue data structure in a programming language of your choice (e.g. C++, Python, Java).
- Analyze the time complexity of the enqueue, dequeue and peek operations using big O notation.

Task 5

Exercise 5:

Title: Analyzing Time Complexity of a Recursive function

Objective: To understand the concept of big O notation and to analyze the time complexity of a recursive function.

Instructions:

- Research the concept of big O notation and its importance in analyzing the time complexity of recursive functions
- Implement a recursive function that performs a specific task (e.g. factorial calculation, fibonacci sequence calculation).
- Analyze the time complexity of your implementation using big O notation.

Deadline

Deadline 3 of March 23.59