

# Final Project: Algorithms and Data Structures - Practical Work

This repository contains the solution to a practical work assignment for the course 'Algorithms and Data Structures'.

The project aims to solve a set of problems related to task allocation and processor scheduling using different algorithmic techniques, including backtracking and greedy algorithms.

The task includes multiple stages:

1. The first stage involves implementing services to process and retrieve information from a set of processors and tasks described in CSV files.

The goal is to create an efficient solution for the following services:

- Service 1: Retrieve all information of a task given its ID.
- Service 2: Generate a list of critical or non-critical tasks as chosen by the user.
- Service 3: Retrieve tasks within a specified range of priority levels.

2. The second stage focuses on task allocation to processors with the objective of minimizing the execution time.

This part considers constraints such as:

- No processor can execute more than 2 critical tasks.
- Non-refrigerated processors cannot allocate tasks exceeding a certain time limit.

The project employs two different algorithmic techniques for task assignment:

- **Backtracking**: The solution will involve generating all possible task assignments and selecting the one with the minimum maximum execution time.
- **Greedy**: The solution will involve choosing tasks based on certain heuristics to minimize

execution time.

Both techniques are evaluated based on:

- The maximum execution time.
- Metrics for analyzing the cost of the solution, such as the number of states generated (backtracking) and the number of candidates considered (greedy).

This repository showcases the solution implemented in Java with a focus on efficiency and algorithmic techniques suitable for real-world task scheduling problems.