Sorting, Binary Trees, Priority Queues and Graphs

Instructions

Sorting

1. Implement the SelectionSort algorithm and present best and worst cases complexity analyisis.
2. Implement the InsertionSort algorithm and present best and worst cases complexity analyisis.
3. Implement the QuickSort algorithm and present best and worst cases complexity analyisis.

Binary Trees

5. Implement the missing methods in the code available at `https://github.com/rcpsilva/PCC104_DesignAndAnalysisOfAlgorithms/blob/main/2025-1/Problem%20sets/Latex%20Source/5_Sorting_BinaryTrees_Heaps_Graphs/binary_search_tree.py`. Present the complexity analyis for each method.

Heaps

Graphs

9. Implement the DFS and the BFS algorithms for a graph represented as adjacency list. Present the space and time complexity analysis in terms of the graph depth, h, and the branching factor (average number of children per node), b.
10. Implement the DFS, the BFS and the A* algorithms to find a path in a maze. The maze is represented as a $m \times n$ matrix, where each cell may contain:
    - `0`: free space;
    - `1`: wall;
    - `s`: starting position;
    - `g`: goal (maze exit).