BAKER COLLEGE
STUDENT LEARNING OUTCOMES

CS3220 Data Structures and Algorithms II
3 Semester Credit Hours

Student Learning Outcomes and Enabling Objectives

1. Apply principles learned in the prerequisite data structure course including, but not limited to implementation of pointers, classes, lists, and linked lists, recursion, stacks, queues, and use of both original and professionally written code
    a. Explain basic Sorting Algorithms
    b. Compare Selection Sort, Bubble Sort, Insertion Sort
    c. Implement Faster Sorting Algorithms, including Merge Sort, Quick Sort, and Radix Sort
2. Use advanced queues, prioritize them, and implement them properly
    a. Implement the ADT Queue, as well as implement the ADT priority queue
    b. Explain Simple Applications of a Queue and read a String of Characters
    c. Implement Position-Oriented and Value-Oriented ADTs
    d. Compare a Link-Based Implementation and an Array-Based Implementation.
3. Implement comparison-based search algorithms including sequential searches, ordered lists, and binary searches to solve problems.
    a. Implement the different types of trees
    b. Explain Full, Complete, Balanced Binary Trees and ADT Trees.
    c. Explain the Maximum and Minimum Heights of a Binary Tree
    d. Explain Binary Tree Operations including Search, Traversals of a Binary Tree, and ADT Binary Search Tree
4. Implement heap functions and analysis to solve problems.
    a. Implement an Array-Based Heap
    b. Implement a heap as a Priority Queue
    c. Explain how the Heap Sort works
5. Explain the balanced tree and the different tree search.
    a. Implement AVL tree
    b. Implement the different tree types, like 2-3 Trees
    c. Implement 2-3-4 Trees
    d. Implement Red-Black Trees
6. Implement graphs and graph algorithms, and associated traversals to solve problems.
    a. Implement Graphs as ADTs and as Traversal
    b. Implement Depth-First and Breadth-First Search
    c. Explain the applications of Graphs
    d. Explain Topological Sorting and Spanning Trees

Big Ideas and Essential Questions

Big Ideas

- Apply advanced concepts associated with algorithm analysis.
- Develop and implement complex algorithms
- Apply principles of Object-Oriented Development and Programming
- Demonstrate the ability to use principles of recursion to solve problems.
- Implementation of advanced programming techniques

Essential Questions

1. How to apply advanced concepts associated with algorithms in writing professional programs to solve problems?
2. How to use advanced hash tables to develop software programs?
3. How to apply the principles of heap, treaps, and priority queues to improve program functionality?
4. How to use principles of recursion to solve problems?
5. How to implement balanced binary search trees?
6. How to research, design, develop, test, run, debug, and implement an advanced-level python program?

These SLOs are approved for experiential credit.

Effective: Spring 2023