Learning Outcomes for Design and Analysis of Algorithms I, ECES 421

This documents the minimum expected of all students completing ECES 421. (The terms ”knowledge,” ”comprehension,” ”application,” ”analysis,” ”synthesis,” and ”evaluation” are here used as in Bloom’s taxonomy.)

1. **Knowledge Goals:**
   (a) The student will demonstrate knowledge of how to measure the complexity of an algorithm, including best-case, worst-case, and average complexities as functions of the input size, as well as classi£cation in terms of asymptotic complexity classes O, Ω, and Θ.
   (b) The student will demonstrate knowledge of correctness proofs of algorithms, including usage of mathematical induction on the input size or to establish loop invariants.
   (c) The student will demonstrate knowledge of basic data structures and their implementations, including lists, arrays, stacks, queues, heaps, trees, and graphs.
   (d) The student will demonstrate knowledge of the basic algorithmic design strategies, including recursion, divide-and-conquer, the greedy method, dynamic programming, and backtracking and branch-and-bound.
   (e) The student will demonstrate knowledge of elementary lower bound arguments for algorithms, including lower bounds for searching and sorting using comparison-based algorithms.
2. **Comprehension Goals:**
   (a) The student will explain search and traversal strategies for rooted trees and graphs.
   (b) The student will explain the implicit stacking involved in recursive function invocations, and trace its action.
   (c) The student will explain why recurrence relations arise naturally when analyzing the complexity of recursive algorithms.
   (d) The student will explain the basic principles behind the major design strategies, including the role played by the principle of optimality in dynamic programming.
   (e) The student will trace the action of a variety of algorithms.
3. **Application Goals:**
   (a) The student will solve recurrence relations for the complexity of a number of recursive algorithms.
   (b) The student will design and give pseudocode for algorithms solving a variety of problems using the major design strategies.
4. **Analysis Goals:**
   (a) The student will solve recurrence relations for the complexity of a number of recursive algorithms, and determine the asymptotic behavior of both recursive and nonrecursive algorithms.
   (b) The student will give correctness proofs for a number of algorithms.
5. **Synthesis Goal:**
   The student will design solutions to problems that require combining several design strategies or data structures in a novel way.
6. **Evaluation Goal:**
   The student will be able to decide between various algorithms solving a given problem. For example, mergesort versus tri-mergesort, mergesort versus quicksort, Kruskal’s algorithm versus Prim’s algorithm, etc.