
Design and Analysis of Algorithms

Across the course, I gained a stronger understanding of how to analyze problems and choose efficient solutions using algorithms and data structures. I learned the importance of evaluating best-case, average-case, and worst-case performance, along with how time and space complexity affect real program design. I also studied core techniques such as brute force, divide-and-conquer, decrease-and-conquer, dynamic programming, and greedy methods, which are useful for solving different kinds of problems. In addition, I learned about graph traversal, shortest-path algorithms, trees like AVL and 2-3 trees, heaps, hashing, and database concepts. Overall, the course strengthened my ability to think more logically about efficiency, structure, and problem-solving in computer science.