### ***Algorithms II***

|  |  |  |
| --- | --- | --- |
| **June 30-July 6**    **Topics**   * Two Motivating Applications (Sequence Alignment and Internet Routing) * Selected Review from Part I (Optional) * Introduction to Greedy Algorithms * A Scheduling Application * Prim's Minimum Spanning Tree Algorithm | **Homework**   * **Due July 13:** * Problem Set #1: Greedy algorithms and MSTs * Programming Assignment #1: Greedy scheduling and Prim's MST algorithm | **Suggested Readings:**   * CLRS: Chapter 16 (Sections 1 and 2) and Chapter 23 * DPV: Sections 5.1.1, 5.1.2, and 5.1.5 * KT: Sections 4.1, 4.2, 4.3, and 4.5 * SW: Section 4.3 |
| **July 7-13**    **Topics**   * Kruskal's Minimum Spanning Tree Algorithm * Clustering * Advanced Topics: On the Union-Find Data Structure * Huffman Codes | **Homework**   * **Due July 20:** * Problem Set #2: More MSTs, and Huffman codes * Programming Assignment #2: Clustering | **Suggested Readings:**   * CLRS Chapter 16 (Section 3), Chapter 21, and Chapter 23 (Section 2) * DPV Sections 5.1.3, 5.1.4, and 5.2 * KT Sections 4.5-4.8 * SW Sections 1.5, 4.3, and 5.5 |
| **July 14-20**    **Topics**   * Dynamic Programming and Applications * The Knapsack Problem * Sequence Alignment * Optimal Search Trees | **Homework**   * **Due July 27:** * Problem Set #3: Dynamic Programming * Programming Assignment #3: The Knapsack Problem | **Suggested Readings:**   * CLRS Chapter 15 * DPV Chapter 6 * KT Sections 6.1-6.6 |
| **July 21-27**    **Topics**   * More Dynamic Programming and Shortest Paths * SIngle-Source Shortest Paths, Revisited * The Bellman-Ford Algorithm * Internet Routing * The All-Pairs Shortest Paths Problem * The Floyd-Warshall Algorithm * Johnson's Algorithm | **Homework**   * **Due August 3:** * Problem Set #4: Shortest Paths * Programming Assignment #4: All-Pairs Shortest Paths | **Suggested Readings:**   * CLRS Section 24.1 and Chpater 25 * DPV Sections 4.6, 4.7, 6.1, 6.6 * KT Sections 6.8-6.10 |
| **July 28-August 3**    **Topics**   * P, NP, and What They Mean * Reductions Between Problems * NP-Complete Problems * The P vs. NP Problem * Solvable Special Cases of NP-Complete Problems * Smarter (But Still Exponential-Time) Search Algorithms for NP-Complete Problems | **Homework**   * **Due August 10:** * Problem Set #5: NP-Complete Problems and Smarter Search Algorithms for Them * Programming Assignment #5: The Traveling Salesman Problem | **Suggested Readings:**   * CLRS Chapter 34 * DPV Section 8.1, 8.2, 9.1 * KT Sections 8.1-8.4, 8.10, 10.1, 10.2 |
| **August 4-10**    **Topics**   * Heuristics with Provable Guarantees * Greedy and Dynamic Programming Heuristics for the Knapsack Problem * Local Search: General Principles, Max Cut, and 2SAT | **Homework**   * **Due August 17:** * Problem Set #6: Approximation Algorithms and Local Search * Programming Assignment #6: 2SAT | **Suggested Readings:**   * CLRS Sections 35.1-35.3 * DPV Section 9.2, 9.3 * KT Sections 11.1-11.3, 11.8, 12.1, 12.4, 12.5 |
| **Final Exam (August 11-24)** |  |  |