南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Algorithm Design and Analysis (H)

CS216

Instructor: Shan CHEN (陈杉)

chens3@sustech.edu.cn

Course Review

Course Chapters

- Introduction
- Algorithm Analysis
- Greedy Algorithms
- Divide and Conquer
- Dynamic Programming
- Intractability
- Network Flow
- Randomized Algorithms

Introduction

- **One-to-one stable matching.**
  - Example: marriage
  - Gale-Sharpley algorithm
  - Perfect stable matching
  - Man optimality vs woman optimality
- **One-to-many stable matching.**
  - Example: medical students applying for hospitals
  - extended Gale-Sharpley algorithm
  - Stable matching
  - Student optimality vs hospital optimality

Algorithm Analysis

- **Computational tractability.**
  - Worst-case/average-case analysis
  - Efficient = worst-case polynomial-time
- **Asymptotic order of growth.**
  - O, Ω, Θ definitions and their properties
  - Common running times: logarithmic, linear, linearithmic, quadratic, cubic, polynomial, exponential, factorial, etc.
- **Five representative problems on independent set.**
  - Interval scheduling, weighted interval scheduling, bipartite matching, independent set, competitive facility location.

Greedy Algorithms

- **Scheduling.**
  - Interval scheduling: greedy algorithm stays ahead
  - Interval partitioning: "structural" bound
  - Scheduling to minimize lateness: exchange argument
- **Graphs.**
  - Single-source/destination shortest paths: Dijkstra's algorithm
  - Single-pair shortest path: A^* search algorithm
  - Minimum spanning trees and k-clustering: Prim's, Kruskal, Borůvka, etc.
  - Min-cost arborescences: Chu-Liu's algorithm and its $O(m \log n)$ implementation
- **Other greedy algorithms.**
  - Huffman codes and optimal caching

Divide and Conquer

- **Divide and conquer.**
  - Divide up problem into several independent subproblems (of the same kind).
  - Solve (conquer) each subproblem recursively.
  - Combine solutions to subproblems into overall solution.
- **Applications.**
  - Merge sort and counting inversions
  - Randomized quick sort and randomized quick select
  - Closest pair of points
  - Integer/matrix multiplication
  - Convolution and the fast Fourier transform (FFT)

Dynamic Programming

- **Dynamic Programming.**
  - Divide up problem into several overlapping subproblems and combine solutions to subproblems into overall solution.
  - Strategy: define subproblems, memorize intermediate results for later use, and order subproblems from "smallest" to "largest".
- **Techniques and applications.**
  - Binary choice: weighted interval scheduling
  - Multiway choice: segmented least squares
  - Adding a new variable: knapsack problem
  - Intervals: RNA secondary structure
  - DP + divide and conquer: sequence alignment
  - Graphs: shortest paths with negative weights, distance vector, negative cycles

Intractability

- **Basic reduction strategies.**
  - Simple equivalence: INDEPENDENT-SET $\equiv_P$ VERTEX-COVER.
  - Special case to general case: VERTEX-COVER $\leq_P$ SET-COVER.
  - Encoding with gadgets: 3-SAT $\leq_P$ INDEPENDENT-SET.
- **Three types of problems.**
  - Decision problem vs search problem vs optimization problem
- **Important complexity classes and examples.**
  - ***P***, ***NP***, ***NP***-complete definitions
  - The first ***NP***-complete problem: CIRCUIT-SAT
  - 3-SAT is ***NP***-complete

Network Flow

- **Theory and algorithms.**
  - Duality: max-flow value = min-cut capacity
  - Ford-Fulkerson algorithm: improve flow value with augmenting paths
  - More advanced algorithms: capacity-scaling, Edmonds-Karp, Dinitz.
- **Applications.**
  - Bipartite matching
  - Disjoint paths
  - Circulation with supplies, demands, and lower bounds
  - Survey design
  - Airline scheduling
  - Image segmentation

Randomized Algorithms

- **Why randomize?**
  - Can lead to simplest, fastest, or only known algorithm for a particular problem.
  - Examples: linear randomized MST algorithms, linear randomized quick select.
- **Applications.**
  - Content resolution
  - Global min cut
  - MAX 3-SAT
  - Load balancing
- **Important math bounds.**
  - Union bound
  - Chernoff bounds