Overview

design and analysis of efficient algorithms

NP- Completeness

proofs

① Structure (recurrence formula)

② design algo, correct, time complexity

③ data structures (CS 240), implementations (pseudo-codes)

Syllabus

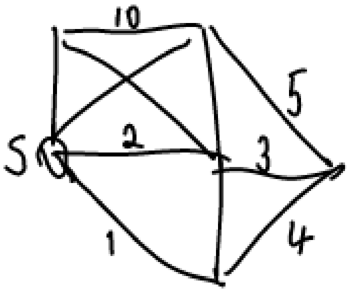
- divide-and-conquer (3 lectures)
- simple graph algos, BFS/DFS (3 lects)
- greedy algos (3-4)
- dynamic programming (4)
- bipartite matching (3)
- NP-completeness (4-5)
- problem sessions (1-2)

---

undirected graph

n-vertices, m-edges

cost on edges



travelling salesman (NP-complete)

visit every vertex ≥ once

min cost tour

chinese postman ($O(n^4)$ w/ matching)

visit every edge ≥ once

min cost tour

n! permutations

10^{10} operations ~ seconds
10^{12} operations ~ minutes
10^{15} operations ~ impossible

2^n time (w/ DP)

Outcome

- well-known algos
- design new algos
- analyse time complexity
- use reductions