Agenda

• NP-Completeness
  - Subset-sum
Updates

• Final on Friday
  - Similar to the midterm
  - Bring your photoID
  - Closed book & closed notes
  - Coverage: From 8/28 lecture to today’s lecture
  - 5 Problems
    • One bonus problem
      - Hard. Unlikely to get partial credit.

• If you want additional help, send me an email directly
Fundamentals

• Know how algorithm works
  - All algorithms mentioned in the lecture
• For graph algorithms, you can assume the graph is connected
Lecture 9

- Variants of shortest-path problems
- Bellman-Ford
  - How it works? When it works? Complexity?
- Single-source shortest path in a DAG
  - Basic idea
- Dijkstra’s algorithm
  - How? When? Complexity?
Lecture 10

• All-pairs shortest path: Floyd-Warshall
  - How? When? Complexity?
  - Dynamic programming: recursive formulation

• Transitive closure
  - Floyd-Warshall (simple way)
  - A better way
  - How? Complexity?
Lecture 11

- Union-Find
  - Operations
  - Representations
    - Linked list representation
    - Disjoint-set forest
    - Difference?
  - Heuristics for union
    - How? Complexity?
Lecture 12

• Dynamic programming
  - Rod cutting
    • Recursive formulation
    • Algorithm: How? Complexity?
  - Chain matrix multiplication
    • Recursive formulation
    • Algorithm: How? Complexity?
Lecture 13

• Dynamic programming
  - Longest common sequence
    • Recursive formulation: formal proof
    • Algorithm: How? Complexity?
    • Similar problems
Lecture 14

• NP-Completeness
  - Definitions of: P, NP, NPC, NP-Hard
  - Relationship
• Reduction $\leq_p$
  - In poly time
  - Yes in A $\iff$ Yes in B
• NP-Completeness proof
  - Show in NP
  - Show it’s NP-Hard
    • Using reduction
Lecture 14

• NP-Complete Problems
  - Directed Hamiltonian cycle
  - Subset-Sum

• Two additional NPC problems
  - Undirected Hamiltonian cycle
  - Set-Partition
The End