Last Time

- NP-Completeness and Reductions.
The Final Exam is **cumulative** with a focus on material from the second half of the semester.

First half:
- Logarithms + Exponents
- Asymptotic Notation
- Recursion
- Inductive Proofs
- Sorting (Comparison, Linear-time)
- Binary Search Trees
- Heaps
- Balanced Trees
Operations on Streams of Data

- Running Mean, Standard Deviation.
- Mean/Std.Dev of last $k$ points.
- Minimum/Maximum
- Selecting 1 element at random
- Selecting $k$ elements at random
Optimization Algorithms

- Dynamic Programming
- Greedy Algorithms
Dynamic Programming

- Optimal Substructure
- Repeated Substructure
- Setting up a DP table for word problems
- Minimum Edit Distance (Levenshtein distance)
- Making Change
Greedy Algorithms

- Proving that a greedy choice leads to an optimal solution
  - Local Solution must be part of the Global Solution
- Fractional Knapsack
- Activity Selection

**Case Study:** Huffman Coding

- Constructing the Huffman Tree
- Calculating the average encoded size of each token
Graphs

- Traversal – BFS, DFS
- Shortest Paths – Dijkstra, Bellman-Ford
- Strongly Connected Components
- Minimum Spanning Trees – Prim’s, Kruskal’s
- Detecting Negative Cycles – Bellman-Ford
Hashing

- Definitions: Hash Table, Hash Function
- Chaining
- Handling collisions
- Open Addressing
- Perfect Hashing
Definitions: NP-Complete, NP-Hard, $P \neq NP$ vs. $P = NP$

Logic and structure of an NP-completeness proof (by reduction)

Reductions to show that a problem is NP-Complete
Definitions: Span, Work, Parallelism

Calculating the above.
Final Review

- Final Exam is 2hrs long 4:00-6:00
- One 8.5 x 11 sheet of notes.
- Where: A223
Bye

- Course Evaluations.
- Next time
  - Final Exam