CSE421: Algorithms

Professor: Anup Rao (anuprao@cs)

Algorithms

- A recipe for carrying out a computation efficiently
- Studied before there were computers
- Eg: Gaussian elimination, gcd, multiplication …
- How to design algorithms?
- How to analyze efficiency and prove correctness?

Approximate Schedule

Weeks 1-2: Introductions, asymptotics, some basic algorithms. Graphs, trees, connectivity, testing bipartiteness.

Weeks 3: Greedy algorithms for Interval Scheduling, Minimum Spanning Tree, Set Cover, Vertex Cover

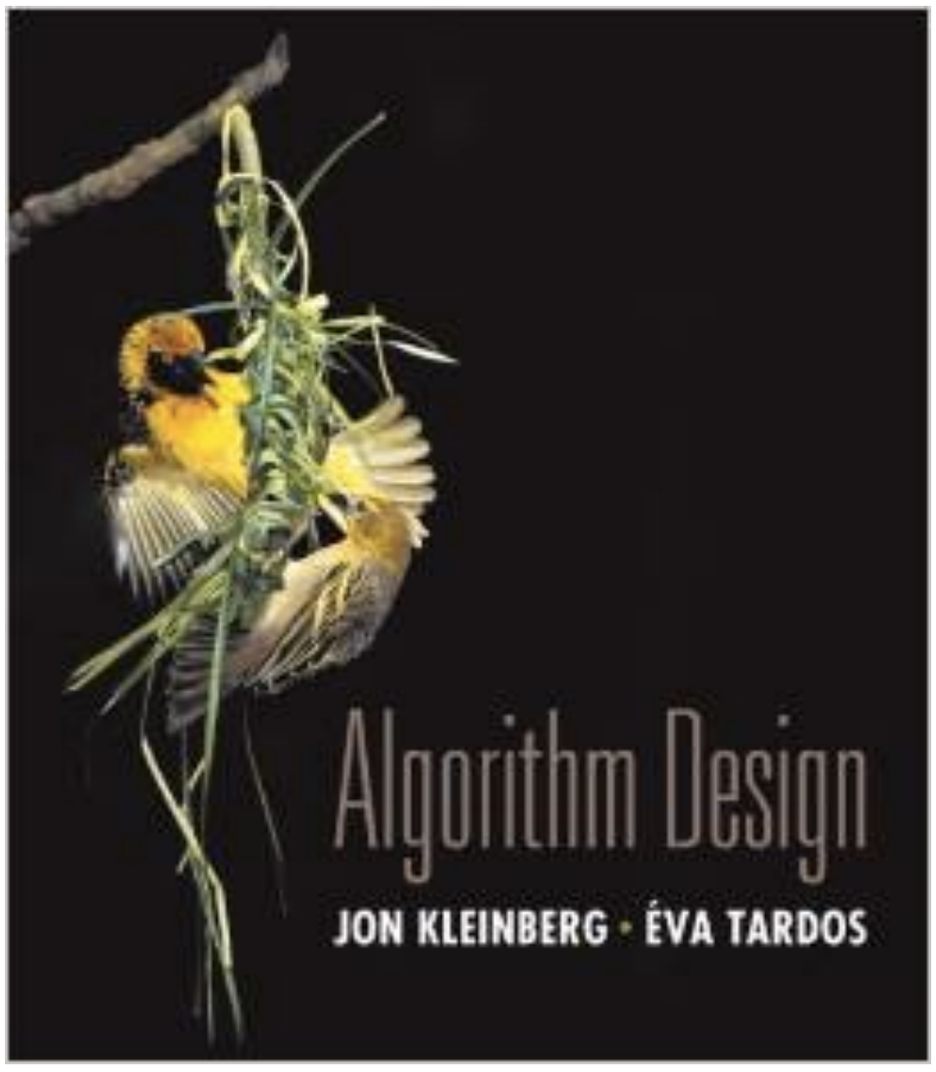
Weeks 4-5: Divide and Conquer. Algorithms for Sorting, Selection, Closest, Multiplication. Midterm

Week 6: Dynamic Programming. Algorithms for Edit Distance, Longest Path, Knapsack.

Week 7: Flows and Cuts. Algorithms for Matching, Graph Partitioning.

Weeks 8-10: Linear programming, Randomized algorithms, NP-completeness.

Text Book



Everything you need to know will be in slides. Videos of lectures will be posted on website.

Evaluation

Final: 35%

- in class, cumulative, cheat sheet

Midterm: 25%

- in class, cheat sheet

Homework every week: 40%

-You must write up solutions by yourself, but you can discuss homework with others.

Time Outside Class

Office Hours:

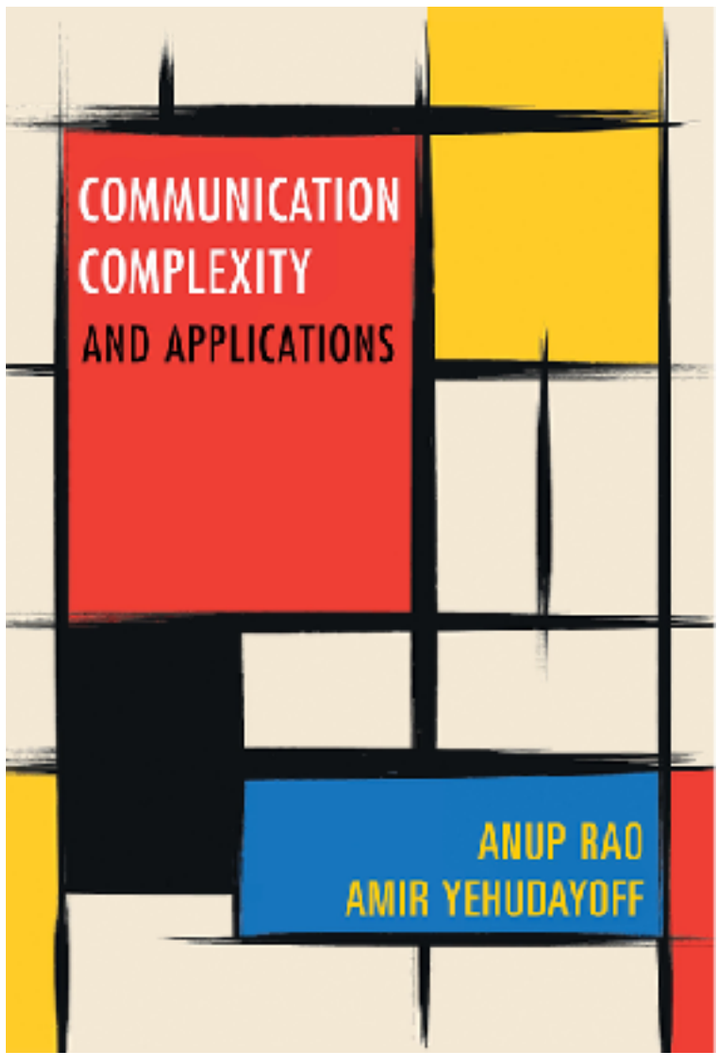
See website for schedule.

To get the most out of this class...

- Start working on the homework early, problems are often deceptively hard.
- Talk to {professor, TAs, students}. Brainstorming sessions really really help!

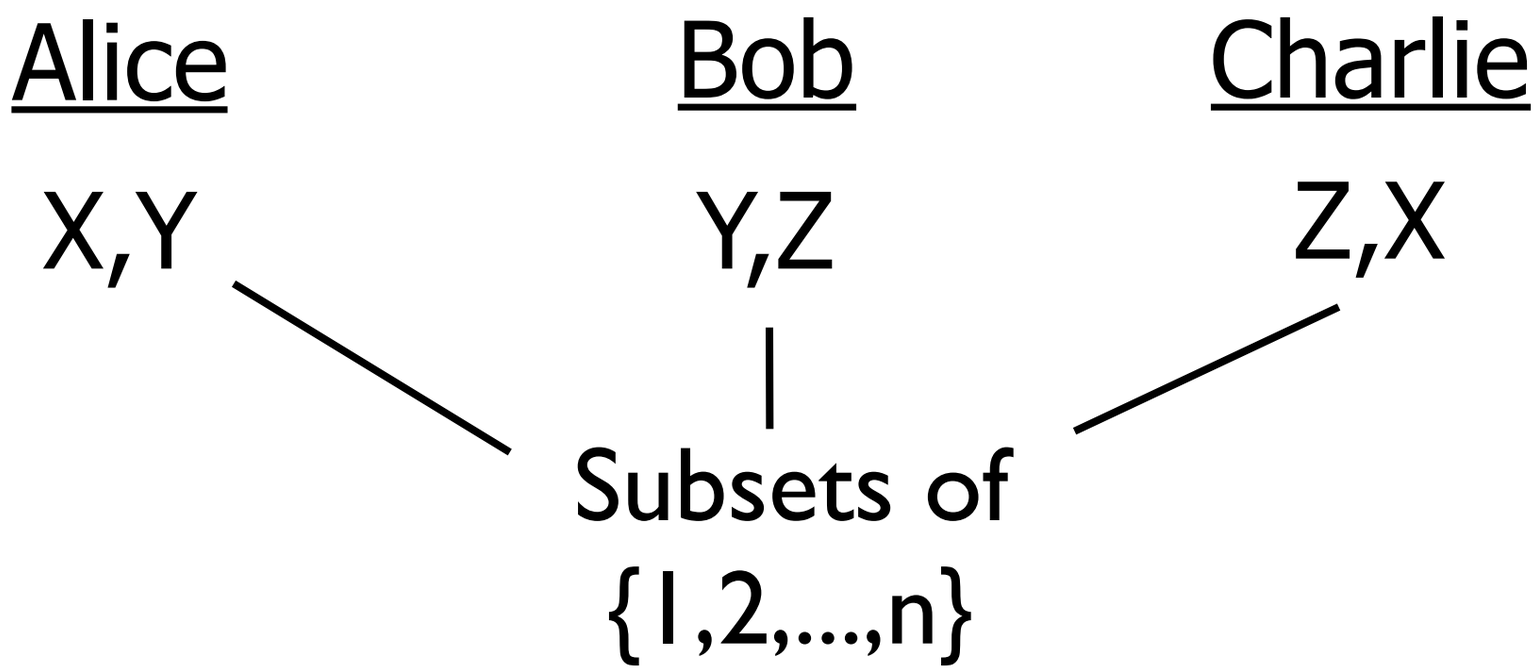
Anup's research: Communication Complexity

If two or more people need to compute a function that depends on all their inputs, how long does their conversation need to be?

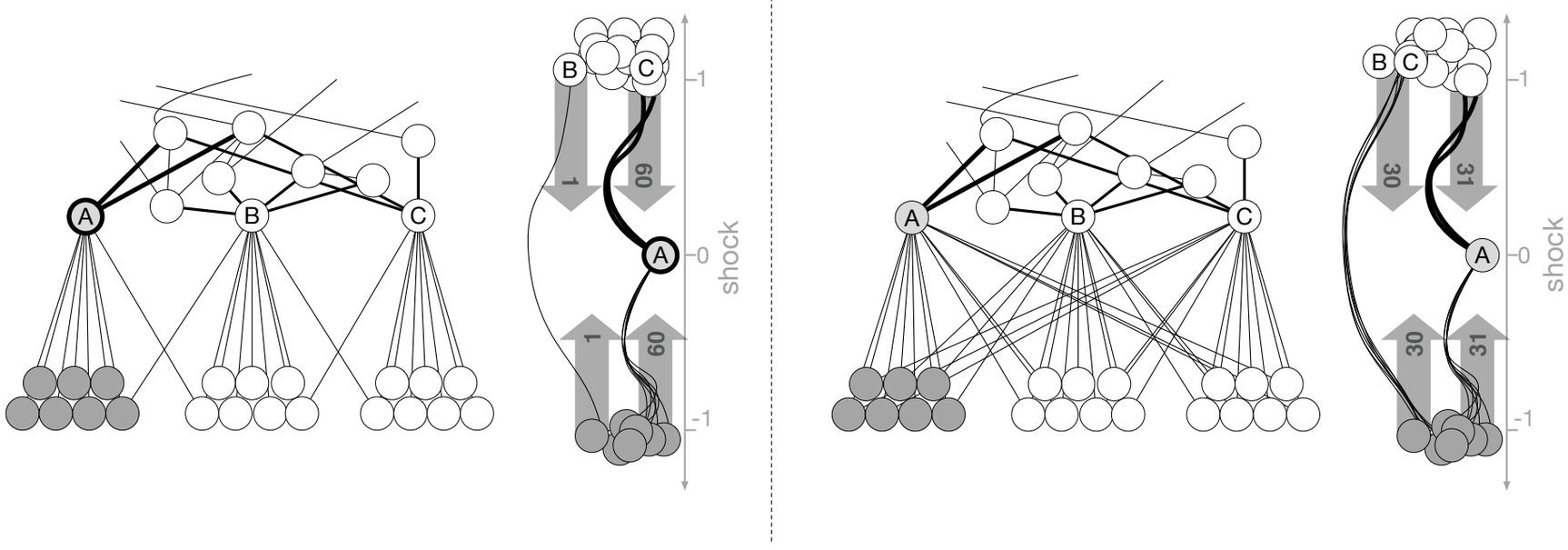


Concrete Problems

How many bits need to be communicated to know if X, Y, Z have a common element?



Economics



A new way to use graph theory to measure the efficiency of an economy.