ANNOUNCEMENTS

▸ Homework 2 out Wednesday

▸ Contacting the TAs: adv-algorithms-ta-fall18@googlegroups.com
LAST CLASS

- Dynamic programming
  - Examples: maximum increasing subsequence – key idea: write recurrence for MIS'(i) – largest increasing subsequence starting with A[i]

- Greedy algorithms
GREEDY ALGORITHMS

▸ Greedy == “myopic”

▸ Algorithm makes sequence of decisions – chooses best at each step, without taking future into account
Matching: suppose we have \( n \) children, \( n \) gifts and “happiness” values \( H_{ij} \). Assign gifts to children to maximize “total happiness”

- Greedy == “myopic”

\( \text{(Greedy can be very bad.)} \)
EXAMPLE

**Scheduling:** given set of jobs with processing times $p_1, p_2, \ldots, p_n$, find an order of processing so that sum of *completion times* is minimized
NATURAL HEURISTIC?

- Do shorter jobs first

  - **Proof:** consider the optimal ordering. If jobs are not in increasing order, swap the first “out of order” pair of neighbors.

- Example of greedy: at each step, think of deciding which job to do

\[
p_{j_1} \leq p_{j_2} \leq \ldots \leq p_{j_n}.\]
GREEDY ALGORITHMS

- Greedy == “myopic”

- Algorithm makes sequence of decisions – chooses best at each step, without taking future into account

- “Building up” solution (easy to design, but hard to reason about).
Spanning tree: suppose we have a weighted undirected graph. Choose subset of edges of minimum total weight that connects all vertices.

In the graph with just the blue edges, we can reach every vertex from every other vertex.
Spanning tree: suppose we have a weighted undirected graph. Choose subset of edges of minimum total weight that connects all vertices.

“Communication backbone” in networks

Set of edges chosen always forms a tree (assuming edge wts are > 0)

connected, a cyclic graph.
GREEDY ALGORITHM — BUILDING UP

**Idea**
Keep track of a set of nodes that we "connected so far". Add one new node to this set in each iteration.

**Implementation of idea**
Look at all edges going out of our set of nodes, add the edge of least weight.
GREEDY ALGORITHM
CORRECTNESS

Guess: use induction.

Statement: let \( u_1, u_2, \ldots \) be the order in which the \( k \)th nodes get connected. Then at each step, the edges chosen so far form a min spanning tree of \( \{ u_1, u_2, \ldots, u_k \} \).
For the set of vertices \( \{ u_1, \ldots, u_6 \} \), the edges chosen form the MST.
CORRECTNESS — INDUCTIVE STATEMENT

Claim. Suppose the algorithm picks edges $e_1, e_2, \ldots, e_k$ in first $k$ steps. There exists an optimal solution that includes all the edges $e_i$. 

i.e., an optimal tree.

for the full graph.

High level argument: 
- Take optimal tree $T$ that includes $e_1, \ldots, e_k$.
- Massage it to also include $e_5$ without increasing cost.
CORRECTNESS — INDUCTIVE STATEMENT

Observation: If you added $e_5$ to $T$, $T \cup \{e_5\}$ has a cycle.

- Removing $e'$ & adding $e_5$ keeps the graph connected.

$$w_t(e_5) \leq w_t(e')$$

\[ = (T \setminus e') \cup \{e_5\} \]

Qn: why don't we know more about $T$?
GREEDY ALGORITHMS

▸ Make decisions in a myopic way – build up solution
▸ Not optimal for many problems, but gives intuition
▸ Proving correctness can be tricky!
MAXIMUM COVERAGE

**Hiring problem:** suppose we have $n$ people, each with a set of skills from some universe $\{1, \ldots, m\}$. Pick $k$ people so as to maximize the total # of distinct skills.

- Each person – set $S_i \subseteq \{1, 2, \ldots, m\}$
- Want to maximize union of chosen sets
GREEDY ALGORITHM — BUILDING UP
OPTIMAL?

- {1,2,3}, {4}, {1,2,4}, {3,5}
CAN GREEDY BE REALLY BAD?
CAN GREEDY BE REALLY BAD?

**Theorem.** “Value” of solution chosen by greedy is at least \((1/2)\) * value of *optimal* solution.

Example of “approximation algorithm”
HOW TO PROVE THIS? FIRST STEP?
KEY OBSERVATION — CAN ALWAYS MAKE “PROGRESS”
INDUCTIVE CLAIM
APPROXIMATION