Goals:
- Design a reduction
- Describe importance of reductions

Reminders/Questions:
- Probability/Quicksort Review: [https://www.cs.middlebury.edu/~skimmel/Courses/302S22/]
- Last day with these groups!
- Why?
- Why work backwards?

Problem: Cell Tower Scheduling

Output: Set of towers to broadcast in the next time step. If no tower with 2 miles of each other broadcast -> failure.

**Cell Tower Theorem (main)**

![Diagram of cell tower scheduling](attachment:image.png)

1. **Physical counters?** Towers with more packets get priority. Geographic inequalities (placing towers fairly) could allow this.

2. Describe combinatorial strategies:
   - Each tower is a vertex
   - Each packet is an edge
   - Weight is # same packets
   - Edges between towers have weight 2 if sum is less than 2 times.

3. What is runtime of each combinatorial strategy?
   - In terms of # of towers.
   - For each step:
     1. Check distance and if
     2. [More details]

More General Reduction

**Reduction F + Reduction Q + Reduction g**

`P (want to solve)`

### **Reduction**

- If `Runtime (I, g) is O(poly(n))`
- `P ∈ Q` means `P` is polynomial time reducible to `Q`
- `P ∈ Q` is harder than `P`
- `Q` gives us the power to solve `P`

**Why think about reductions?**