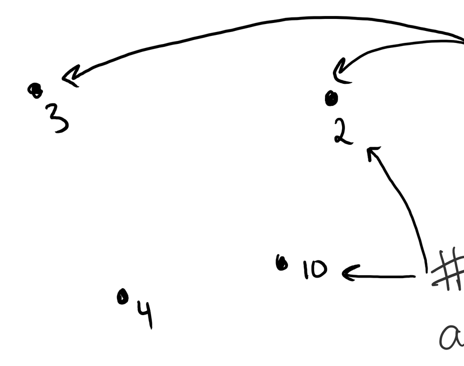
# 5: Reductions

Friday, March 19, 2021 10:35 AM

**Problem:** Cell Tower Scheduling

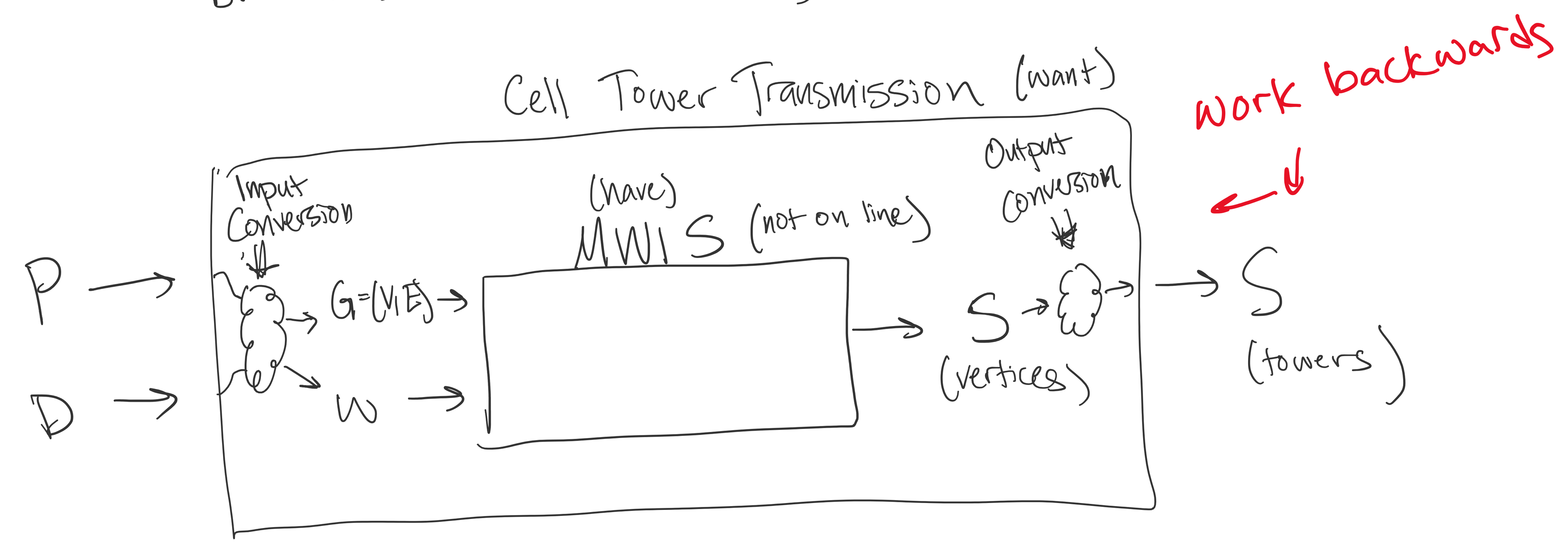



Positions of Towers : P = (position)

# of packets of data queued at tower waiting to transmit : D (data)

**Output:** Set of towers to broadcast in next time step

(If 2 towers w/in 2 miles of each other broadcast → interference.)




1. Ethical concerns?

2. Describe conversion strategies:

   P, D ⇒ V, E, w

   S (vertices) ⇒ S (towers)

3. What is runtime of each conversion strategy? (In terms of n, number of towers.) (Brute force OK)

1. 911 calls, "Rural" users disadvantaged

2. $V = \{1, 2, \dots, n\}$ ← # of towers

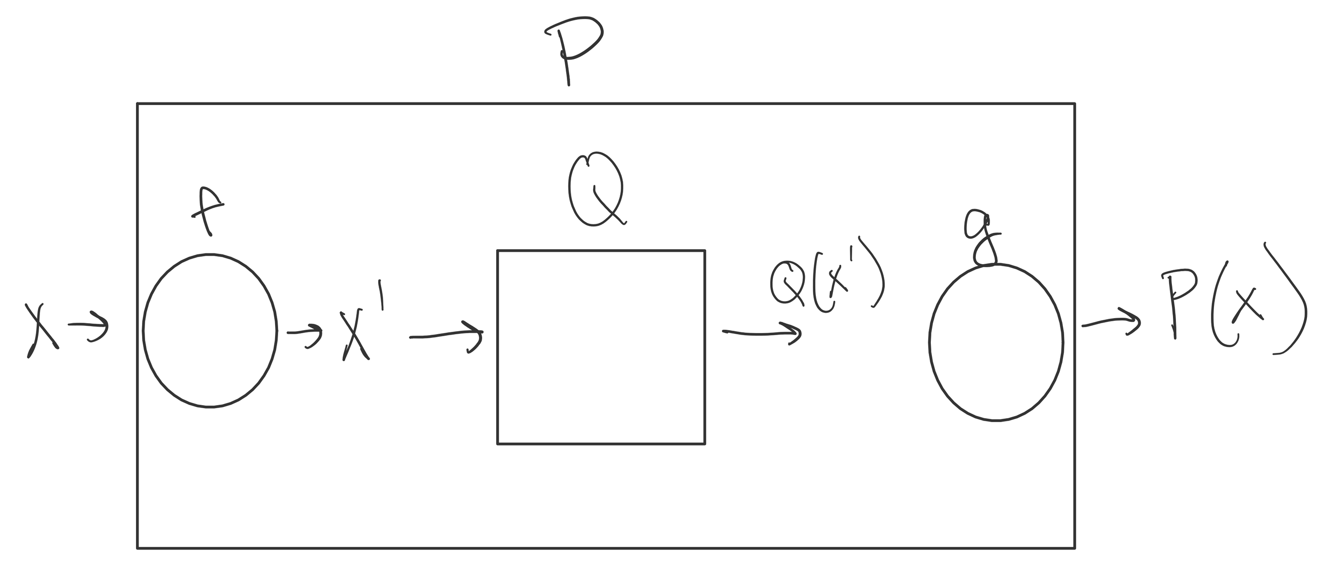
   $E = \{\{u,v\} : d(P[u], P[v]) \leq 2\}$

   $w(u) = D(u)$

   S (vertices) = S (towers)

3. Creating E: $O(n^2)$ time (check each pair)

# More General Reduction




Runtime(P): Runtime(f) + Runtime(Q) + Runtime(g)

Usually want: small, bigger (doing the work)

ex: MWIS not on line $O(2^n)$ so $O(n)$ vs $O(n^3)$ is insignificant

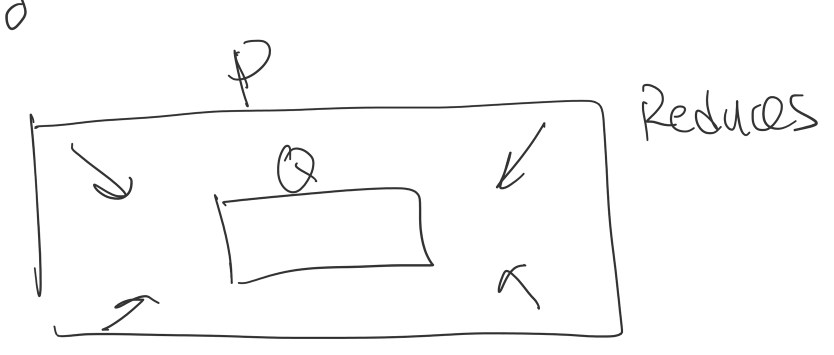
If Runtime(f, g) is O(poly(n)) ← P's input size

$\underbrace{O(1), O(n), O(n^2), O(n^3), \dots O(n^{100}), \dots}_{\text{polynomial time}}$

we say:

"P is polynomial time reducible to Q"

"There is a polynomial time reduction from P to Q"




"P reduces to Q"

We write:

$P \leq_p Q$ b/c Q is "more powerful" than P.

With Q, can solve P or Q

With P, might not be able to solve Q

(confusing b/c P box bigger than Q box)

Why think about reductions?

- Practical: If have alg. for Q, can use to create alg. for P
- Conceptual: Gives us a way to compare "power", difficulty of problems

  e.g. Halting Problem