Last Time

• Reductions
  – Solve one problem by using code from another
    • Element Uniqueness ≤ Sorting
    • Matrix Multiplication ≤ Matrix Squaring
    • Matrix Squaring ≤ Matrix Multiplication
    • NewProblem ≤ Traveling Salesman
    • Traveling Salesman ≤ NewProblem
  – P/NP/NP-Complete
    • PATH/HAMPATH
CNF Satisfiability

• **Input:** Boolean variables in conjunctive normal form
• **Output:** True/false values for each variable that would made the full statement true

• _Example Input:_
  - \((x) \land (\bar{x} \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor y \lor z \lor w) \land (\bar{x} \lor y)\)

• _Example Output:_
  - \(x = T, y = T, z = T, w = F\)
3-CNF Satisfiability

- **Input:** Boolean variables in conjunctive normal form, such that each clause contains exactly 3 literals
- **Output:** True/false values for each variable that would make the full statement true

- **Example Input:**
  \[(x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor y \lor z)\]

- **Example Output:**
  \[x = F, y = T, z = F\]
CNF Reductions

• Are CNF-SAT and 3SAT in the same complexity class?
  – Need to show CNF-SAT ≤ 3SAT and 3SAT ≤ CNF-SAT

• 3SAT ≤ CNF-SAT ?
  – Trivial

program 3SAT(i)
    return CNF-SAT(i)
CNF Reductions

• CNF-SAT ≤ 3SAT ?

program CNF-SAT(i)
    Read formula i
    Transform formula i into formula j
    return 3SAT(j)
CNF Reductions

• CNF-SAT $\leq$ 3SAT?
  – Does the clause have exactly 3 literals?
    • Done
  – Does the clause have less than 3 literals?
    • Add extra literals that mutually cancel out
    • $(\bar{x} \lor y) \rightarrow (\bar{x} \lor y \lor A) \land (\bar{x} \lor y \lor \bar{A})$
  – Does the clause have more than 3 literals?
    • Break the clause up into a collection of clauses that have linked canceling variables
    • $(\bar{x} \lor y \lor z \lor \bar{w} \lor q) \rightarrow (\bar{x} \lor y \lor B) \land (\bar{B} \lor z \lor C) \land (\bar{C} \lor \bar{w} \lor q)$
Vertex Cover

• **Input:** Graph G, integer K
• **Output:** 1 if G has a vertex cover of size k
  0 otherwise

• **Vertex Cover** – A collection S of vertices such that every edge is adjacent to at least one vertex in S

\[
\begin{array}{c|c}
  k & \text{Output} \\
  \hline 
  3 & 1 \\
  2 & 0 \\
\end{array}
\]
NP-Hard

• A problem X is **NP-Hard** if there exists an NP-Complete problem Y such that $Y \leq X$ in polynomial time (written as $Y \leq_p X$).

• **Theorem:** Vertex Cover is NP-Hard

• **Proof:** $3SAT \leq VERTEX-COVER$

```plaintext
program 3SAT(i)
    Read formula i
    Transform formula i into graph G
    Pick k (such that G has a vertex cover of size k if and only if i is satisfiable)
    return VERTEX-COVER(G,k)
```
VERTEX-COVER

• Variable Gadget

• Clause Gadget
Figure 4: Example graph from 3-SAT to k-Vertex Cover reduction. The boolean formula for this example is $\phi = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor \neg x_2 \lor \geq x_3)$. The variable and clause gadgets for this boolean formula are visible in this graph, as well as the edges connecting nodes from the literals in the variable gadgets to matching literals in the clause gadgets.

• In this case, we pick
  – $k = \text{num\_variables} + 2 \times \text{num\_clauses}$

• Does a vertex cover of size $k$ imply a satisfiable formula?
  – Pick the vertices from the variable gadget

• Does a satisfiable formula imply a vertex cover of size $k$?
  – Pick the vertices from the variable gadget and fill in the rest
Any Questions?