Lectures and Reading. Next week will continue reductions showing NP-hardness from the ALR notes and Arora-Barak. Friday may get to diagonalization from Arora-Barak chapter 3.

(1) A short-essay topic

We refer to the proof of the Cook-Levin Theorem that was given in class. Suppose we were to negate every variable in every block of three clauses for incoming wires $u, v$ and outgoing wire $w$ from a gate $g$, so that it now reads:

$$(\bar{u} \lor \bar{w}) \land (\bar{v} \lor \bar{w}) \land (u \lor v \lor w).$$

But suppose we still require that the output wire $w_0$ outputs 1 for acceptance. Is the construction still correct—or correctable? And please show exactly what would happen if we were to use the other universal gate, NOR, in place of NAND in the construction of $\phi_n$ in the proof. (18 pts.)

Answer: Given a binary string $x$, define $x'$ to be the result of flipping each bit of $x$, e.g., $10110' = 01001$. Let $A$ be the language we are reducing to 3SAT and $R(x, y)$ the witness predicate that goes with $A$ being in $\textbf{NP}$, with $M_R$ as the polynomial-time verifier. If we were to negate every literal on every wire, then the original proof would apply to the languages

$$L''_R = \{\langle x', y' \rangle : \neg R(x, y)\}.$$

Note the negation in $L''_R$, which comes about because of the negation of the output wire $w_0$. But the proof is not correct when we insist on the original $x$ and $y$ and on $w_0 = 1$ for acceptance. We could fix the $x$ and $y$ part with extra NOT gates but they are not provided for by the proof, and that still leaves the $w_0$ issue.

It is “correctable” by noting that $\textbf{P}$ (deterministic polynomial time) is closed under complements, and both $\textbf{P}$ and $\textbf{NP}$ are closed under the $A$-to-$A'$ operation, which when repeated flips back to $A$. Given any language $A \in \textbf{NP}$, we can work the body with the flipped-and-negated witness predicate

$$R''(x, y) \equiv \neg R(x', y').$$

The proof still doesn’t work literally in the body when you convert $R''$ into NAND gates, however. What you get is the AND of negated inputs, which is not universal. Instead the correction just goes back to saying that the original proof with all literals negated is working for $R''$.

The better way is to use the fact that NOR is also a universal gate. The key observation is that

$$\text{NOR}(u, v) = w \iff (\bar{u} \lor \bar{w}) \land (\bar{v} \lor \bar{w}) \land (u \lor v \lor w).$$

Thus when we stick with the original $R(x, y)$ but build the circuit out of NOR instead of NAND gates, we get exactly the formula as-described with the original (not flipped) $x$ and $y$ inputs and $w_0 = 1$ as the acceptance condition.
Show that the problem of deciding whether an undirected graph is connected belongs to polynomial time. Sketch an algorithm in pseudocode and estimate its asymptotic running time in terms of the number \( n \) of vertices and number \( m \) of edges. (Note that this could be part of a witness predicate for the “Unbreakable” problem on Assignment 5. 18 pts.)

**Answer:** Do breadth-first search from any given vertex \( s \):

```java
set<Node> FOUND = {s};
set<Node> NOVEL := {s};
int n = |V|;
int count = 0;
while (!empty(NOVEL)) {
    for (each u \in NOVEL) {
        for (each v such that (u,v) \in E) {
            if (v \notin FOUND) {
                FOUND += {v};
                NOVEL += {v};
                count++;
            }
        }
    }
    NOVEL -= {u};
}
return (count == n);
```

The outer two loops go through every found vertex exactly once. The inner loop goes through up to \( n-1 \) remaining nodes. Hence if the set operations can be done with random access then the time is \( \tilde{O}(n^2) \). Another way of looking at the algorithm is that it traces each edge in the component of \( s \) exactly once. Then the time is \( \tilde{O}(m) \), which of course equals \( \tilde{O}(n^2) \) when \( m = \Theta(n^2) \).

Without random access on Turing machines we may need to square this time, but anyway it remains polynomial in \( n = |V| \) and \( m = |E| \) for the graph \( G = (V,E) \). With random access, it is arguably linear in the size of the graph, which is \( \tilde{O}(m) \) when the graph is given as an edge list. One can also solve this by spanning-tree algorithms, which run in similar times.

(3) Choosing the New Administration.

The President-elect needs to fill a fairly large number of federal positions—many more besides the Cabinet—from an even larger number \( n \) of qualified people. He/she also listens to various advisors and special-interest groups, who submit a large number \( m \) of positive and negative recommendations. A positive recommendation lists 1 or 2 or 3 or more people and says, “choose at least one of these guys.” A negative recommendation gives a similar list but says, “Don’t choose all of these guys—that would be favoring this group too much.” The recommendations don’t necessarily specify a particular federal position—they only talk about whether particular people \( x, y, \) and/or \( z \) should belong to the Administration in general.

Consider the task of choosing whether each person should belong to the Administration. Can this be done in a way that meets every recommendation? Prove that the decision problem framed by this question is NP-complete, preferably by reduction from (3)SAT and in any event.
by showing how this problem is “SAT-like.” (Thus in a formal, computational sense it is **hard** to form a government. 21 pts., for 57 on the set)

**Answer:** The input to the problem is the list $V$ of people and the list $E$ of recommendations, and we can take $n + m$ as its size. Since a choice of people for the Administration is a subset $S$ of $V$, and since checking that $S$ satisfies all the recommendations is simple to do, the existence of such an $S$ defines a language in **NP**—call it $L_A$.

By the given proof of the Cook-Levin Theorem, 3SAT remains **NP**-complete even when restricted to instances $\phi(x_1, \ldots, x_n) = C_1 \land C_2 \land \cdots \land C_m$ in which every clause $C_j$ has all-positive or all-negative literals. Call this restricted **NP**-complete problem 3SAT$'$ . To reduce 3SAT$'$ further to the stated problem, let the “people” be $x_1, \ldots, x_n$, and let a truth assignment $x_i = 1$ mean that person $x_i$ is selected to be in the Administration. Then the all-positive clauses $C_j$ become the positive recommendations, and the all-negative clauses become the negative recommendations. The function $f$ mapping $\phi$ to the corresponding case of the Administration-forming task is kind-of an identity function, and hence computable in polynomial (indeed linear) time. Since $\phi \in 3\text{SAT}' \iff f(\phi) \in L_A$, $L_A$ is **NP**-complete.

**Technote:** When the number $k$ of available positions to fill is fixed and specified, the problem is **NP**-complete even without the negative recommendations and with positive recommendations of size 2! Indeed it becomes the **Vertex Cover** problem: turn each person into a vertex, and each recommendation into an edge. So that was an alternative answer.