15-451 Algorithms, Fall 2017
Recitation 6 Worksheet

I. Exam Questions. Ask them now.

II. 2-Color It's a well known fact that graph 3-Colorability is NP-complete. Fortunately, however, 2-COLOR is in P. In fact, it is $O(n)$. How can we use DFS to find if a graph is 2-colorable?

III. 2-SAT It's also a well known fact that 3-SAT (Given a boolean formula of the form $(x_1 \vee \bar{x_2} \vee x_3) \wedge (\bar{x_3} \vee \bar{x_1} \vee x_4) \wedge ...$, determining if there is a satisfying assignment) is NP-complete. Also fortunately, however, 2-SAT is in P. In fact, it too is $O(n)$, though perhaps this fact is less obvious than the case for 2-COLOR.

1. Consider constructing the implication graph of a 2-SAT formula. We want the property a directed edge represents an implication; i.e. a directed edge (u, v) implies that if u is true, then v must be true. How would you go about doing this (Hint: for each variable, we want 2 vertices corresponding to x_i and $\bar{x_i}$)

2. Show that if there is some x_i and $\bar{x_i}$ are strongly connected, then the formula is unsatisfiable.

3. Show that for each strongly connected component, either all of the predicates in the component are true or false.

4. Consider the DAG of strongly connected components. Assuming x_i and $\bar{x_i}$ are always in different components, come up with a 'coloring' technique for the DAG such that we never get a $T \to F$ edge (Hint: consider the sources/sinks of this graph). This implies the inverse of (2), meaning the formula is satisfiable iff $\forall x_i$, x_i and $\bar{x_i}$ are in different components.

5. Verify this indeed runs in linear time