Homework 7
Writing session on Wednesday October 25

---

1. **(SOLO)** A language $L \subseteq \{0,1\}^*$ is called *skinny* if there is some constant $k \geq 0$ such that for all $n \in \mathbb{N}$, we have $|L \cap \{0,1\}^n| \leq n^k$. Show that any skinny language can be computed by a polynomial-size circuit family. (Hint: Exercise 10.13 from the notes may be relevant.)

2. **(SOLO)** Let $H \subseteq \{0,1\}^*$ consist of all strings containing at least two 1's. Describe a circuit family of size $O(n)$ computing H (a proof that your circuit family correctly computes H is not required, but it should be clear from your construction that it is indeed correct). Prove that the size bound is as required. (Hint: Think recursive.)

3. **(SOLO)**

   (a) Let DOUBLE-CIRCUIT-SAT be the following problem: given a Boolean circuit, are there two or more assignments to the input gates that make the circuit evaluate to 1? Show that CIRCUIT-SAT polynomial-time reduces to DOUBLE-CIRCUIT-SAT.

   (b) Let 4SAT be the following problem: given a Boolean formula in conjunctive normal form in which every clause has exactly 4 literals, is there a truth assignment to the variables that makes the formula evaluate to true? Show that 3SAT polynomial-time reduces to 4SAT.

   (c) Suppose there are n people $\{p_1, p_2, \ldots, p_n\}$ and k activities $\{a_1, a_2, \ldots, a_k\}$. Each person p_i is associated with a subset of activities $L_i \subseteq \{a_1, a_2, \ldots, a_k\}$ which she wishes to participate in. Moreover, we are given a natural number m. We would like to figure out whether it is possible to schedule all k activities in m different time slots such that no person has two activities with a time conflict. Let's denote the corresponding decision problem by S. Show that 3COL polynomial-time reduces to S.

4. **(GROUP)** Consider the following game played on a connected undirected graph G:

   (i) Initially Player 1 chooses a vertex u to start on.

   (ii) Player 2 chooses a vertex v neighboring u and then deletes u and all the edges incident to u from the graph.

   (iii) Player 1 chooses a vertex w neighboring v and then deletes v and all the edges incident to v from the graph.

   (iv) etc.

   Note that a chosen vertex is removed from the graph so that it cannot be rechosen in subsequent rounds. A player loses the game if, on their turn, they are not able to pick a vertex.

   (a) Show that if G contains a perfect matching, then there is a winning strategy for player 2 (i.e., there is a strategy for player 2 that ensures that he always wins the game no matter what player 1 chooses to do).

   (b) Show that if G does not contain a perfect matching, then there is a winning strategy for player 1.

5. **(GROUP**) In the regular Stable Matching problem, we assumed that participants prefer to be matched over being single. In this question, we will remove this condition, as it may not be always a realistic assumption. We modify the original Stable Matching problem so that everyone now has a list of preferences which can potentially be incomplete (i.e., the list may not include every possible person from the opposite set of participants). As before, a person prefers being matched with someone on their list over being single. But a person prefers being single over being matched with someone not on their list. In this modification, the notion of a *stable matching* is a matching (not-necessarily perfect) in which:

   - There is no pair (m, w) where m and w are not matched to each other, but they prefer each other over their current situation (i.e., their current match or being single).
   - There is no matched person who prefers being single over being matched with their current partner.

   (a) Does a stable matching always exist in this setting? Prove your answer.

   (b) Is it true that if a person is single in one stable matching, then he or she is single in all stable matchings (i.e., is it true that there is no hope for some people)? Prove your answer.

6. **(OPEN)** Define an *orientation* of G to be a directed graph obtained by assigning a single direction to each edge of G. In this question, we are going to prove the following statement:

   > The minimal number of colors k such that G is k-colorable is exactly the largest number for which every orientation of G contains a directed path with k vertices.

   To prove this, we use the following sub-steps:

   (a) Suppose that G is k-colorable. Give an orientation of G which does not contain a directed path with $k + 1$ vertices.

   (b) Suppose there is an orientation G' of G such that: (i) G' contains no directed cycles, and (ii) any simple directed path in G' consists of at most k vertices. Show that G is k-colorable.

   (c) Suppose there is an orientation of G in which any directed path consists of at most k vertices. Show that G is k-colorable.

   (d) Conclude the main statement.

   For fun, read an application of graph coloring to compiler optimization: `http://www.lighterra.com/papers/graphcoloring/`.