CSE531: Complexity Theory February 25, 2022

Homework 3

Anup Rao Due: March 6, 2022

Read the fine print[1]. Each problem is worth 10 points:

1. Consider the following game between two players: Given a directed graph $G = (V, E)$, and a start vertex s, the players (starting with Player 1) alternately choose an outgoing edge incident to the current vertex to reach a vertex that was not previously visited. If one of the players cannot choose a next vertex, he loses. Let $\mathbf{GAME}(G)$ be the function that is 1 if and only if Player 1 has a strategy that ensures that she always wins no matter what Player 2 does.

   Show that **GAME** is in **PSPACE**.

2. In class we showed that the expected time for the randomized 2SAT algorithm to find a satisfying assignment, if one exists, is $O(n^2)$. Here we show that the same analysis does not help to show that the algorithm works for 3SAT.

   The Chernoff bound states that $X_1, \ldots, X_n$ are independent random variables such that

   $$X_i = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1 - p, \end{cases}$$

   then for every ϵ, $\Pr[\sum_{i=1}^{n} X_i > pn(1 + \epsilon)] < 2^{-\epsilon^2 pn/4}$.

   (a) Assume that the 3SAT formula has exactly one satisfying assignment, and this assignment has distance at least $n/2$ from the initial assignment that the algorithm starts out with (this is the hard case for the algorithm). Argue that the only way that the algorithm can succeed is if there is some contiguous interval of $n/2$ steps, where in at least $n/4$ of those steps, the algorithm moves towards the satisfying solution. To do this, consider the final $n/2$ steps in a sequence of steps that leads the algorithm towards a satisfying solution.

   (b) Use the Chernoff bound to argue that at any point during the run of the algorithm, the probability that the next $n/2$ steps will have $n/4$ steps that move towards the satisfying solution is exponentially small. Use the union bound to conclude that the probability that the algorithm finds the solution in time $2^{o(n)}$ is at most $2^{-\Omega(n)}$.

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.

(c) Finally, use linearity of expectation to argue that the expected time for the algorithm to find a solution is $2^{O(n)}$. To do this, observe that at any point if the algorithm makes n moves towards the satisfying solution, then it will find the satisfying solution. Compute the probability that this happens, and use linearity of expectation to compute the expected time for this to happen. Show that this means that the probability the algorithm does not find the solution in $2^{O(n)}$ time is at most $2^{-\Omega(n)}$, by Markov's inequality.

3. Suppose **TQBF** is also **PSPACE**-complete under log-space reductions—meaning that for every $f \in \mathbf{PSPACE}$, there is a logspace computable function h such that $f(x) = \mathbf{TQBF}(h(x))$. Prove that this implies that $\mathbf{TQBF} \notin \mathbf{NL}$. Hint: Use Savitch's theorem and one of the hierarchy theorems.