CSE431: Complexity Theory	October 4, 2018
Homework 1	
Anup Rao	Due: October 12, 2018

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

1. Show that every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a branching program with $O(2^n/n)$ nodes.

2. Show that there is a constant c such that for n large enough, there is a function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by a branching program with less than $\frac{2^n}{cn}$ nodes.

3. Someone invents a new kind of circuit called a *large* fan-in circuit. In a large fan-in circuit, each (non-input) gate can take as input as many as $n/2$ bits from other gates, and then compute any function of those $n/2$ bits. So a gate is labeled by any function $h : \{0,1\}^{n/2} \to \{0,1\}$. Use a counting argument to show that for n large enough, there is a function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by large fan-in circuit of size at most $2^{n/3}$.

4. In class we discussed the difficulty (and appeal) of showing that every small circuit can be simulated by a small-depth circuit. Here we explore how to prove this for *formulas*. A formula is a circuit where every gate has out-degree at most 1. In other words, each gate is used as an input for at most 1 gate. (We are allowed to duplicate gates that correspond to input variables in circuits, so input variables can be read more than once in a formula.)

   [0 points] Can you explain why such a circuit is called a 'formula'?

   In this problem, we shall show that if $f : \{0,1\}^n \to \{0,1\}$ can be computed with a formula of size s, then it can be computed by a formula of depth $O(\log s)$.

   Since we are interested in showing that depth is asymptotically bounded by $O(\log s)$, it is enough to handle the case when $s > 6$, say.

   For every gate g of the formula, let $c(g)$ be the number of gates (including g) used to compute g in the formula. First, we show that there must be a gate g in the formula such that $s/3 \leq c(g) \leq 2s/3 + 1$. Let us define a sequence of gates $g_1, g_2, \ldots$ as follows. Start by setting g_1 to be the output gate of the formula, so $c(g_1) = s$. For each i, if g_i is an input gate (so $(c(g_i) = 1)$, we end the sequence of gates. Otherwise, if h, q are the two gates that feed into g_i, we set g_{i+1} to be the gate that maximizes $c(g_{i+1})$. Since we have $c(g_i) = c(h) + c(q) + 1$, this choice ensures that $c(g_{i+1}) \geq (c(g_i) - 1)/2$.

[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`.

The idea is that every gate in the sequence has at least half as many gates as the previous one, so some gate of the sequence must have $c(g)$ in the desired interval. To prove this, let i be the largest index with $c(g_i) \geq 2s/3 + 1 > 5$, by the assumption that $s > 6$. g_i cannot be an input gate, since $c(g_i) > 1$. We claim that $s/3 \leq c(g_{i+1}) \leq 2s/3 + 1$. This is because $c(g_{i+1}) \leq 2s/3$ by the choice of i, yet we also have

$$c(g_{i+1}) \geq (c(g_i) - 1)/2 \geq (2s/3 + 1 - 1)/2 = s/3.$$

This completes the first part of the proof.

Next, complete the proof by using the gate g found above to construct three new formulas: one that computes the value of g, one that computes the value of f when $g = 1$, and one that computes the value of f when $g = 0$. Inductively apply the same process to make each of these formulas of small depth, and combine their outputs to obtain the final formula. Carry out the calculations to show that the resulting formula is of depth $O(\log s)$.