Warm-up Exercises

1. Let $f, g : \mathbb{N} \to \mathbb{N}$ be two functions. Recall that $f = O(g)$ if there exists a $c > 0$ such that $f(n) \leq c \cdot g(n)$ for every sufficiently large n. We say that $f = \Omega(g)$ if $g = O(f)$ and that $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$. Also, we say that $f = o(g)$ if for any $\varepsilon > 0$, $f(n) \leq \varepsilon \cdot g(n)$ for every sufficiently large n. Finally, we say that $f = \omega(g)$ if $g = o(f)$.

   Prove or disprove:

   (a) $(5n)! = O(n!^5)$.
   (b) If $f(n) = O(n)$ then $10^{f(n)} = O(2^n)$.
   (c) $\log(n!) = \Theta(n \log n)$.
   (d) Every two functions f, g satisfy $f = O(g)$ or $g = O(f)$.
   (e) There exists a function f such that $f(n) = O(n^{1+\varepsilon})$ for any $\varepsilon > 0$ but $f(n) = \omega(n)$.

2. (a) Show that all Boolean functions can be expressed in terms of 'and' and 'not'.
   (b) Define the connectives NOR and NAND as follows:

   $$\mathsf{NOR}(x, y) = \neg(x \vee y) \text{ and } \mathsf{NAND}(x, y) = \neg(x \wedge y).$$

   Show that all Boolean functions can be expressed in term of NOR alone.
   Repeat for NAND.
   (c) Show that there is no other single binary connective besides these two that can express all Boolean functions.

3. Prove that there exists a Turing machine that is a circuit simulator. That is, a Turing machine that given an n-input circuit C and an input $x \in \{0,1\}^n$ outputs $C(x)$.
   Explain how you choose to represent a circuit and analyze the running time.

Exercises for Submission

1. The n-input parity function $\mathsf{parity}_n : \{0,1\}^n \to \{0,1\}$ is defined to output the number of input bits that equal one, modulo 2.

   (a) Show that parity_n can be computed with circuits of size $O(n \cdot 2^{\sqrt{n}})$ and constant depth. The gates may have large fan-in.
   (b) Show that parity_n can be computed with circuits of size $O(n)$ and depth $O(\log n)$.

2. (a) Define the function $\mathsf{add}_n : \{0,1\}^{2n} \to \{0,1\}^{n+1}$ that calculates the sum of two n bit binary integers and produces the $n+1$ bit result. Show that the add_n function can be computed with $O(n)$ size circuits.
   (b) Define the function $\mathsf{majority}_n : \{0,1\}^n \to \{0,1\}$ by $\mathsf{majority}_n(x_1, \ldots, x_n) = 0$ if $\sum x_i < n/2$ and 1 otherwise. Thus the $\mathsf{majority}_n$ function returns the majority vote of the inputs. Show that $\mathsf{majority}_n$ can be computed with $O(n \log n)$ size circuits.
   Hint: Use (2a). Bonus: Improve it to $O(n)$.

3. We define a partial order on $\{0,1\}^n$ by $x \leq y$ if for every $i \in \{1, 2, \ldots, n\}$, $x_i \leq y_i$. We say that $f : \{0,1\}^* \to \{0,1\}^*$ is monotone if whenever $x \leq y$ we have $f(x) \leq f(y)$. A circuit is monotone if it has only 'and' and 'or' gates. Prove that a monotone circuit computes a monotone function, and that every monotone nonconstant function has a monotone circuit.

4. Prove that $\mathsf{SIZE}(n)$ is strictly contained in $\mathsf{SIZE}(n^2)$ (i.e., $\mathsf{SIZE}(n) \subsetneq \mathsf{SIZE}(n^2)$).