<u>Problem set 1</u>

Computational Complexity.

1. Let M be a k-string TM and f be a function from the nonnegative integers to the nonnegative integers. We say that M *computes* f if for every input 1^n, M halts with $1^{f(n)}$ on its last tape. That is, given a unary representation of a number n, it outputs the unary representation of $f(n)$.

   We say that f is a *proper complexity function*, if f is a nondecreasing function from the nonnegative integers to the nonnegative integers and there exists a k-strings TM M that computes f, such that, for every n, M halts in time $O(n + f(n))$ and uses space $O(f(n))$.

   (a) Show that if f and g are proper complexity functions then so are $f(g)$, $f + g$, $f \cdot g$ and 2^g. You may assume that $f(n) \geq n$.

   (b) Show that the following are proper complexity functions: $\log^2 n, n^2, 2^n, \sqrt{n}$.

2. Let C be a class of functions from nonnegative integers to nonnegative integers. We say that C is closed under *left polynomial composition* if $f(n) \in C$ implies $p(f(n)) = O(g(n))$ for some $g(n) \in C$, for all polynomials $p(n)$. We say that C is closed under *right polynomial composition* if $f(n) \in C$ implies $f(p(n)) = O(g(n))$ for some $g(n) \in C$, for all polynomials $p(n)$.

   For each of the following classes of functions determine whether it is closed under left polynomial composition and whether it is closed under right polynomial composition.

   (a) $\{n^k \mid k > 0\}$

   (b) $\{k \cdot n \mid k > 0\}$

   (c) $\{k^n \mid k > 0\}$

   (d) $\{2^{n^k} \mid k > 0\}$

   (e) $\{\log^k n \mid k > 0\}$

   (f) $\{k \cdot \log n \mid k > 0\}$

3. Each string of an $f(n)$ time bounded k-strings TM M is divided into blocks of size $\sqrt{f(n)}$. The machine is *block respecting* if the block boundaries are crossed only in steps that are integer multiple of $\sqrt{f(n)}$.

   Show that for every $f(n)$ time bounded k-strings TM M there exists an equivalent $O(f(n))$ time bounded $O(k)$-strings TM M' which is block respecting.

4. Describe a Boolean circuit, consisted of $\neg, \vee, \wedge$ gates and n input gates, that calculates the parity function. The size of the circuit should be $O(n)$ and its depth $O(\log n)$. Recall that:

   - The parity of n Boolean variables is the number of variables that equal one, modulo 2.
   - The size of a circuit is the number of edges it contains. The depth of a circuit is the longest directed path from an input gate to an output gate.

5. Let OptCircuit be the following problem:
   Input: A Boolean circuit C.
   Question: Is there a circuit C' of smaller size which is equivalent to C, that is, it calculates the same Boolean function C does.

   Show that $OptCircuit \in PSPACE$.