Problem 1

Recall that a search algorithm for a search problem R outputs, on input x, a string y such that $(x, y) \in R$ if such a y exists.

Consider the search problem R defined as follows:

> $((M, x, z, 1^t), y) \in R$ if $|y| \leq t$, z is a prefix of y, and M is a deterministic Turing Machine that accepts input (x, y) in at most t steps.

(a) Show that if $L_R \in \mathrm{P}$, then there is a polynomial-time search algorithm for R.

(b) Show that if P = NP, then every NP-search problem has a polynomial-time search algorithm.[1]

Problem 2

Let R be an NP-search problem. Show that there exists a search algorithm A for R with the following properties.

- For every (not necessarily efficient) search algorithm M for R and every input $x \in L$, if M on input x halts within t steps, then A on input x halts within $p_M(|x|, t)$ steps, where p_M is some polynomial whose coefficients may depend on the description of M but not on x or t.
- For every $x \notin L$, A on input x halts within $2^{|x|^{O(1)}}$ steps.

Hint: Try running different Turing Machines on input x.

[1] The same argument also shows that if $\mathrm{NP} \subseteq \mathrm{P/poly}$, then every NP-search problem can be solved by a circuit family of polynomial size.

Problem 3

Let $s(n)$ be a function such that $s(n) = o(2^n/n)$.

(a) Show that there exists a language L and a string x such that $L \in \mathrm{SIZE}(s(n))$, but $L \cup \{x\} \notin \mathrm{SIZE}(s(n))$.

(b) Using part (a), show that $\mathrm{SIZE}(s(n)) \neq \mathrm{SIZE}(s(n) + O(n))$.

(c) Why can't we use the same argument to "prove" that $\mathrm{DTIME}(n^3) \neq \mathrm{DTIME}(n^3 + O(n))$?

Problem 4

In this problem we prove circuit lower bounds for the polynomial hierarchy.

(a) Show that $\Sigma_4 \not\subseteq \mathrm{SIZE}(n^{10})$.

(b) Show that $\Sigma_2 \not\subseteq \mathrm{SIZE}(n^{10})$. (Use part (a).)