1. Let MAXSIZE be the set of all sets $A \subseteq \{0,1\}^*$ such that for every circuit family $\langle C_n \rangle_{n \in \mathbb{N}}$ which computes $A$, every set $B_n \subseteq \{0,1\}^n$ can be computed by a circuit of size at most $|C_n|$. Note that by Shannon’s theorem, $|C_n| \geq 2^n/n$, for $n \geq 2$.

Recall that the complexity class $E = \text{DTIME}(2^{O(n)}) = \bigcup_c \text{DTIME}(2^{cn})$.

(a) Show that there is $A \in E^{PH}$ such that $A \in \text{MAXSIZE}$.

Hint: See the proof of Kannan’s Theorem given in class.

Solution:
For each $n \in \mathbb{N}$ let $s_n$ be the largest $s$ such that there is a circuit $C_n$ of size $s$ with $n$ inputs and one output such that $C_n$ is not equivalent to any smaller circuit.

Note that using DNF (disjunctive normal form) we know $s_n = O(n2^n) = O(2^{2n})$.

Define $A$ as follows: $x \in A$ iff $C(x) = 1$, where $C$ is the lexicographically first circuit of size $s_n$ with $|x|$ inputs and one output which is not equivalent to any smaller circuit.

Then by definition $A \in \text{MAXSIZE}$.

Let $B = \{\langle x, y \rangle : x \in A$ and $|y| = 2^{|x|}\}$. Then $B \in PH$, because the condition $|y| = 2^{|x|}$ can be checked in time polynomial in $(|x|, |y|)$, and the condition $x \in A$ above can be expressed by a polytime relation preceeded by quantifiers bounded in length by a polynomial in $2^{|x|}$.

To show that $A \in E^{PH}$ it suffices to show that $A \in E^B$. In fact a Turing machine $M$ with oracle $B$ can compute $A$ as follows: On input $x$, $M$ writes $\langle x, y \rangle$ on its oracle tape, where $y$ is a string of 1’s of length $2^{|x|}$. Then $M$ accepts $x$ iff the oracle $B$ returns YES. Note that $M$ runs in time $O(2^{|x|})$.

(b) Show that if $\text{NP} \subseteq \text{P/poly}$ then there is $A \in \text{E}^{\Sigma_2^p}$ such that $A \in \text{MAXSIZE}$.

Solution:
If $\text{NP} \subseteq \text{P/poly}$ then by the Karp/Lipton Theorem the polynomial hierarchy $\text{PH}$ collapses to $\Sigma_2^p$. So (b) follows from (a).

(c) Show that if $\text{P} = \text{NP}$ then there is $A \in \text{E}$ such that $A \in \text{MAXSIZE}$.

Solution:
If $\text{P} = \text{NP}$ then $\text{PH} = \text{P}$, so by (a) it follows that the set $A$ defined in part (a) is in $\text{E}^P$. But we can show that $\text{E}^P = \text{E}$ as follows.

Let $M^C$ be an oracle TM computing $A$, where $C \in \text{P}$, and $M$ runs in time $O(2^{cn})$. We may assume that some TM computes $C$ in time $O(n^k)$, for some $k$. Then on input $x$ of length $n$, each oracle query made by $M^C$ has length $O(2^{cn})$, and can be answered in time $O(2^{cn}) = O(2^{ckn})$. Further $M^C$ can make only $O(2^{cn})$ such queries. Hence a TM with no oracle can simulate $M^C$ in time $O(2^{ckn} \cdot 2^{cn}) = O(2^{dn})$ for some constant $d$. 


2. (See Exercise 6.13 in text). A Boolean formula can be viewed as a Boolean circuit in which every node (except the input nodes) has out-degree 1. Show that a language is computed by a polynomial size family of formulas iff it is in $\text{NC}^1$.

**Hint:** The graph underlying the formula (after inputs are deleted) is a rooted tree in which each node has at most two children (its inputs). Show that every such tree with $m \geq 2$ leaves has a subtree with between $m/3$ and $2m/3$ leaves.

**Solution:**
First we show that every language in $\text{NC}^1$ can be computed by a polynomial size family of formulas. A simple induction on $d$ shows that every Boolean circuit of depth $d$ is equivalent to a formula of size at most $O(2^d)$. For the induction step, note that the output gate of the circuit has fan-in at most 2, and we can apply the induction hypothesis to each input gate.

The $n$-th circuit $C_n$ in an $\text{NC}^1$ circuit family has depth $O(\log n)$, so the corresponding formula has size $2^{O(\log n)} = n^{O(1)}$ (polynomial size).

For the converse, we need to prove the Hint. Let $T$ be a binary rooted tree with $m \geq 2$ leaves. Starting at the root of $T$, and walk toward the leaves, always choosing the subtree with at least half of the number of leaves of the current subtree. Stop when reaching a subtree $T'$ with at most $2m/3$ leaves. Then $T'$ has at least $m/3$ leaves since the previous subtree has more than $2m/3$ leaves. Thus $T'$ is the desired subtree.

Now we show that a polynomial size family of formulas can be converted to an equivalent circuit family of depth $O(\log s)$. It suffices to show how to convert a formula $\varphi$ with $s$ leaves to an equivalent circuit of depth $O(\log s)$. (We may assume that all nodes have fan-in 2, by pushing not gates to the leaves, using DeMorgan’s laws.)

Now we prove by induction on $s \geq 4$, that a formula $\varphi$ with $s$ leaves is equivalent to a formula $\varphi'$ with depth at most $C \log_2 s$, where the constant $C$ will be determined. If $s \leq 4$, let $\varphi' = \varphi$. Otherwise apply the Hint to the tree structure of $\varphi$ to obtain a subformula $\psi$ with between $s/3$ and $2s/3$ leaves. Let $\hat{\varphi}(y)$ be $\varphi$ with the subformula $\psi$ replaced by a new variable $y$. Thus $\varphi$ is the same as $\hat{\varphi}(\psi)$, and $\varphi$ is equivalent to $\varphi_1$, which is given by

$$\varphi_1 = (\psi \land \hat{\varphi}(1)) \lor (\neg \psi \land \hat{\varphi}(0))$$

Note that $\hat{\varphi}(1)$ and $\hat{\varphi}(0)$ each have at most $2s/3$ leaves which are variables, and in general we may replace a formula involving the constants 0 and 1 and variables with an equivalent formula of no greater depth without the constants.

Finally let $\varphi'$ be $\varphi_1$ with the equivalents of subformulas $\psi$, $\hat{\varphi}(1)$, and $\hat{\varphi}(0)$ replaced by equivalent small depth formulas given by the induction hypothesis. Thus the depth of $\varphi'$ is at most $C \log_2((2/3)s) + 3$. This is at most $C \log_2 s$, provided $C \geq 3/\log_2(3/2)$.

3. Do problem 7.1, page 141 in the text. (One can efficiently simulate choosing a random number from 1 to $N$ using coin tosses.)

**Solution:**
Given $N \geq 1$ and $\delta$ with $0 < \delta < 1$ we are to give a probabilistic algorithm $A$ running in $\text{poly}(\log N \log(1/\delta))$ time with output in $\{1, \ldots, N, ?\}$ such that
(1) conditioned on not outputting ?, A’s output is uniformly distributed in $[N]$, and
(2) the probability that $A$ outputs ? is at most $\delta$.

Let $n$ be the smallest integer such that $2^n \geq N$. Repeat $\lceil \log(1/\delta) \rceil$ times:

Choose a random number $t$ between 1 and $2^n$ by choosing $n$ random bits. If $t \leq N$ then output $t$ and halt.

Output ?

4. Do Problem 7.10, page 142 in the text. (The random walk idea for showing connectivity does not work for directed graphs.)

**Solution:**
Let $G_n$ be the graph on vertices $\{1, \ldots, n\}$ in which $s = 1$, $t = n$, there is an edge from $i$ to $i + 1$ for $1 \leq i < n$, and an edge from $i$ to 1 for $1 < i \leq n$. (Note that $G_n$ is strongly connected).

Then the probability of reaching $t$ from $s$ without returning to $s$ is $2^{2-n}$.

Let $E$ be the expected number of returns to $s$ before reaching $t$. Then either there are no returns, or there is at least one return, so

$$E = 2^{2-n} \cdot 0 + (1 - 2^{2-n}) \cdot (1 + E)$$

Solving this equation gives $E = 2^{n-2} - 1$. Hence the expected number of edges traversed before reaching $t$ is $\Omega(2^n)$. 
