1. Show that for any time-constructible $T : \mathbb{N} \to \mathbb{N}$ there is a universal nondeterministic TM $\mathcal{U}$ such that if a nondeterministic TM $M$ runs in time $T(n)$ then $\mathcal{U}$ on input $x$ and $[M]$ runs in time $O(T(n))$ and outputs $M(x)$. (Note that this is more efficient than the case of deterministic TMs. The algorithm is also much simpler.) Hint: Check the computation one work tape at a time.

2. Show that if $A, B \in \text{NP}$ then $A \cap B$ and $A \cup B$ are in $\text{NP}$.

3. Define $\text{M2SAT}=\{[\phi, k] \mid \phi$ is a 2CNF formula for which there is an assignment satisfying at least $k$ clauses of $\phi\}$. Prove that $\text{M2SAT}$ is $\text{NP}$-complete. Hint: It is OK to have the same clause appear more than once.

4. Two Boolean formulas are equivalent iff they have the same set of variables and they compute the same Boolean function. A Boolean formula is minimal if no smaller Boolean formula is equivalent to it. Let $\text{MIN-FORMULA}$ be the set of minimal Boolean formulas. Show that if $P = \text{NP}$ then $\text{MIN-FORMULA} \in P$.

5. A Boolean decision tree is a rooted binary tree in which each internal node is labelled by an input variable $x_i$ and the two outedges of a node are labelled 0 and 1 respectively and each leaf node is labelled 0 or 1. We can associate a Boolean function $f_u$ with each node $u$ of the tree:

- for a leaf, it is the value of the leaf
- for a non-leaf node $u$ labelled $x_i$,

$$f_u = (\overline{x_i} \land f_{u_0}) \lor (x_i \land f_{u_1})$$

where $u_0$ and $u_1$ are the children reached by following the 0 and 1 edges respectively.

The function computed by the tree is the function associated with the root of the tree. Such a tree is oblivious if the variables labelling the vertices on each root-leaf path are in the same order.

(a) Show that every Boolean function $f \in \mathbb{B}_n$ has a Boolean formula of size $O(2^n)$ and depth $O(n)$ by simulating oblivious Boolean decision trees.
(b) At the lowest levels of the Boolean decision trees for part (a), many of the functions being computed will be the same. Use this idea, together a method to simultaneously compute all possible functions on a set of $k$ bits in order to build Boolean circuits of size $O(2^n/n)$ for arbitrary Boolean functions $f \in \mathbb{B}_n$.

6. (Extra credit) Prove that almost all Boolean functions $f \in \mathbb{B}_n$ have formula size $\Omega(2^n/\log n)$ and depth at least $n - O(\log \log n)$. 