There are 2 questions for a total of 10 points.

---

1. (5 points) Prove that the language

$$\{x \mid x \text{ is the binary representation of } n!, \text{ without leading 0's, for some } n \in \mathbb{N}\}$$

is not regular.

**Solution:** Consider the set $S$ of binary representations of $(2^n - 1)!$ for all $n$ in $\mathbb{N}$.

We claim that all of these are in different equivalence classes.

Consider any two distinct natural numbers $n, m$, and wlog assume $n < m$. Suppose $(2^n - 1)!$ has a binary representation $x$, and $(2^m - 1)!$ has a binary representation $y$.

Consider the binary representation of $(2^n - 1)! \times (2^n)$ and $(2^m - 1)! \times (2^n)$. Clearly, they are $x$ followed by $n$ zeros and $y$ followed by $n$ zeros respectively.

The first of them is the binary representation of $(2^n)!$, and hence is in the language.

Now note that

$$(2^m - 1)! < (2^m - 1)! \times 2^n < (2^m - 1)! \times 2^m = (2^m)!$$

Since this implies that this integer lies strictly between two consecutive factorials, the fact that factorials form a monotonically non-decreasing sequence implies that it is not the factorial of any integer. Hence, its binary representation is not in the language.

So one of $x0^n$ and $y0^n$ is in the language, while the other one is not, so both of them are in different equivalence classes.

This shows that any two strings in the set S are in different equivalence classes. Since $S$ is infinite, there are infinitely many equivalence classes.

Since there are infinitely many equivalence classes, the language can't be regular, as needed.

2. (5 points) Let $L_k \subseteq \{0,1\}^*$ be the language defined as

$$L_k = \{xy \mid x, y \in \{0,1\}^k, \text{ and the bitwise-AND of } x \text{ and } y \text{ is } 0^k\}.$$

Observe that $L_k$ is finite, and hence, regular. Prove that for all $k$, any DFA that recognizes $L_k$ has at least $2^k$ states.

**Solution:** Using the Myhill Nerode theorem, it suffices to show that there are at least $2^k$ equivalence classes of $=_{L_k}$, since the number of states in the minimal DFA equals the number of equivalence classes of $=_{L_k}$.

For this, we shall show that each binary string of size $k$ is in its own equivalence class.

Consider any two distinct binary strings $x$ and $y$. We say that a string $u$ is dominated by string $v$ (and equivalently, $v$ dominates $u$) iff for all $i$, $u[i] = 1 \implies v[i] = 1$.

We make two cases:

1. $x$ does not dominate $y$.

   Let $x'$ be the bitwise NOT of $x$ (i.e., $x'[i] = 0$ if and only if $x[i] = 1$). Then $xx'$ is in $L_k$ since the bitwise AND of $x$ and $x'$ is $0^k$ by definition of bitwise NOT. Now since $x$ does not

dominate $y$, there must exist an $i$ such that $x[i] = 0$ and $y[i] = 1$. Then we have $x'[i] = 1$, so the bitwise AND of $y$ and $x'$ has a 1 at the $i^{\text{th}}$ bit, which means that $yx'$ is not in $L_k$. Since we have exhibited $x'$ such that exactly one of $xx'$ and $yx'$ is in the language and the other isn't, this shows that $x$ and $y$ belong to different equivalence classes of $=_{L_k}$.

2. $x$ dominates $y$.

   Let $y'$ be the bitwise NOT of $y$. Since $x$ dominates $y$, $y[i] = 1 \implies x[i] = 1$. If the converse were true (i.e., if it were the case that for all $i$, $x[i] = 1 \implies y[i] = 1$), then it would imply that $y = x$, which doesn't hold for unequal $x$ and $y$. Hence, there exists at least one $i$ for which the converse is false, i.e., for which $x[i] = 1$ and $y[i] = 0$. In this case, $x[i] = 1$ and $y'[i] = 1$, so $xy'$ is not in $L_k$. Also, by a similar argument as in the previous case, $yy'$ is in $L_k$. Hence, $x$ and $y$ are in different equivalence classes of $=_{L_k}$ again.

Since the above cases are exhaustive, we can see that each binary string of length $k$ belongs to a distinct equivalence class of $=_{L_k}$, and from here, it follows that there are at least $2^k$ equivalence classes of $=_{L_k}$. Combining this with the first comment, we are done.