1. Consider the following problem: we are given a parameter k, n red strings $r_1, \ldots, r_n$ and n blue strings $b_1, \ldots, b_n$, and want to find i and j such that the Hamming distance between r_i and b_j is at most k. The strings are binary and of length $d = \omega(\log n)$. Show that, assuming SETH, this cannot be solved in $\mathcal{O}(n^{2-\epsilon})$ time, for any $\epsilon > 0$.

(2 points) 2. Consider two strings $a[1..n]$ and $b[1..m]$. Show how to construct in linear time two new strings $a'[1..N]$ and $b'[1..M]$ such that $N = \mathcal{O}(n)$, $M = \mathcal{O}(m)$, and the edit distance between a and b can be obtained with a simple formula from the LCS of a' and b'.

3. Design an algorithm that computes the LCS of $a[1..n]$ and $b[1..m]$ in $\mathcal{O}((n + m^2) \log n)$ time.

(2 points) 4. Consider two Boolean matrices $A[1..N][1..N]$ and $B[1..N][1..N]$ and let $C[1..n][1..n]$ be their (Boolean) product. Show how to construct a text $t[1..n]$ and a pattern $p[1..m]$ (over a possibly large alphabet) with n and m being $\mathcal{O}(N^2)$ such that every $C[i][j]$ can be obtained from the number of mismatches between some $t[i..(i + m - 1)]$ and $p[1..m]$. What does this say about the complexity of pattern matching with mismatches?