Exercise 1 (Using fingerprints to check matrix multiplication). [★]
The best known algorithms for multiplying two \( n \times n \)-matrices take time \( O(n^\alpha) \) for some \( \alpha > 2 \). However, there is a simple randomized algorithm for verifying matrix products with high probability in \( O(n^2) \) time.

Suppose I claim that \( AB = C \) where \( A, B, \) and \( C \) are integer-valued \( n \times n \)-matrices. You can confirm this by applying both sides of this equation to a random vector \( v \), and checking that \( ABv = Cv \).

\[ \text{Question 1.1} \]
Describe how to check if \( ABv = Cv \) in \( O(n^2) \) time, i.e. without computing the \( n \times n \)-matrix product \( AB \).

We desire to improve furthermore the checking procedure by using fingerprints. Assume that \( AB \neq C \) and let \( \alpha = \max_{i,j} \{|A_{ij}|, |B_{ij}|, |C_{ij}|\} \) bound the maximum coefficient in \( A, B \) and \( C \).

\[ \text{Question 1.2} \]
Explain how to choose a prime number \( p = O(\log(n\alpha) \log \log(n\alpha)) \) such that with probability at least \( \frac{9}{10} \), we have \( AB - C \neq 0 \mod p \).

\( \text{Hint.} \) Use that the \( k \)th prime number \( p_k \) verifies \( 0.91 k \ln k < p_k < 1.7 k \ln k \), as proved by Felgner in 1990.

Assume now that we are lucky and \( AB - C \neq 0 \mod p \).

\[ \text{Question 1.3} \]
Show that if \( v \) is chosen uniformly at random from \( \{0, 1, \ldots, p - 1\}^n \), then the probability that \( ABv = Cv \mod p \) holds is at most \( 1/p \).

Exercise 2 (FPT algorithm for spotting \( k \) disjoint triangles). [★]
Given \( G = (V, E) \) an undirected graph \( (n = |V| \) and \( m = |E|) \) and \( k \) an integer, we are looking for \( k \) vertex-disjoint triangles in \( G \). Note that this problem is \( NP \)-complete when \( k \) is part of the input. We are looking for an algorithm of time complexity \( O(f(k)n^a m^b) \) where the exponents \( a \) and \( b \) are constant, independent of \( k \). Such an algorithm is called \( FPT \) for Fixed Parameter Tractable, which means that the complexity is a fixed-degree polynomial in the size of the input for any fixed value of the parameter \( k \). Consider the following randomized algorithm:

\[ \text{Algorithm 1} \] FPT randomized algorithm for \( k \) disjoint triangles

- Choose independently for each vertex \( u \) a color \( c_u \in \{1, \ldots, 3k\} \) uniformly at random.
- \textbf{return} “Yes” if there is a colorful solution, i.e. a set of \( k \) triangles whose \( 3k \) vertices use exactly once each color; \textbf{return} “I don’t know” otherwise.
Question 2.1) Show that if $G$ contains $k$ disjoint triangles, then the probability that the algorithm answer “Yes” is at least $e^{-3k}$.

Hint. use that: $k! \geq (k/e)^k$ for all $k$.

Question 2.2) How many times should you run this algorithm to improve success probability to $1/2$?

In order to check whether a colorful solution exists, we propose to try all permutations $\pi$ on $\{1, \ldots, 3k\}$ and check if there is any triangles of colors $(\pi_1, \pi_2, \pi_3), \ldots, (\pi_{3k-2}, \pi_{3k-1}, \pi_{3k})$.

Question 2.3) Describe an algorithm that decides if such a collection of triangles exists. (Explicit the exact data structure you are using). What is the overall expected time complexity in $k$, $n$, and $m$, of the algorithm that uses this method to return $k$ disjoint triangles with probability at least $1/2$ if they exist in $G$? What is the time complexity if $k$ is fixed?

Exercise 3 (Binomial laws composition). Show that when the laws considered are independent, $\text{Bin}(\text{Bin}(n, p), q) \sim \text{Bin}(n, pq)$ for all $n \in \mathbb{N}$ and $p, q \in [0, 1]$.

Exercise 4 (Prüfer sequence). Consider a (unrooted) tree with $n$ nodes labelled from 1 to $n$. The Prüfer sequence encoding this tree is obtained by removing vertices from the tree until only two vertices remain. Specifically, at step $i$, remove the leaf with the smallest label and set the $i$th element of the Prüfer sequence to the label of this leaf’s neighbor. For instance, the Prüfer sequence for the tree below is $2, 2, 5, 5, 7, 7, 7$.

\[ \begin{array}{cccccccc}
1 & 2 & 7 & 4 \\
3 & 9 & 7 & 6 \\
5 & & & 3
\end{array} \]

Question 4.1) Show that every labelled unrooted tree corresponds a unique sequence in $\{1, \ldots, n\}^{n-2}$ and reciprocally.

Hint. First, show how to compute the degree of every node from a given Prüfer sequence.

Question 4.2) Conclude that there are $n^{n-2}$ labelled unrooted trees with $n$ nodes.

Question 4.3) More precisely, show that there are

\[
\binom{n-2}{d_1-1, d_2-1, \ldots, d_n-1} = \frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}
\]

labelled unrooted trees with $n$ nodes whose $i$th vertex has degree $d_i$.  

2