1. **Longest path via *color coding.*** In this task, we want to solve the LONGEST PATH problem: given a graph G and an integer k, find a simple path on at least k vertices in G. We want to develop a parameterized algorithm with parameter k.

   First we solve a restriction of the problem called COLORED LONGEST PATH. In addition to the input of LONGEST PATH, the vertices of the graph are pre-colored with one of k colors $\{1, \ldots, k\}$ where this k is the same k as in LONGEST PATH (meaning that there is a coloring $c: V(G) \to \{1, \ldots, k\}$ on input). And the solution path we want to find has to have exactly one vertex from each color class.

   Now suppose that we can solve COLORED LONGEST PATH. We will convince ourselves later that a LONGEST PATH instance $\mathcal{I} = (G, k)$ is a yes-instance if and only if there exists a k-coloring $c: V(G) \to \{1, \ldots, k\}$ of G.

   The question is, how do we find such a coloring? We will find it randomly, meaning that we will color each vertex of G independently, uniformly at random with one of colors from $\{1, \ldots, k\}$. We will convince ourselves later that the random coloring has the property from the preceding paragraph that we need with probability p (which will be small).

   So if we want an algorithm with success probability p, then we need to color the graph uniformly randomly, and then solve an instance of COLORED LONGEST PATH. But in computer science, we want the algorithm to have at least constant success rate (or polynomial rate). How do we go from success rate of p to $\Omega(1)$ (or $\Omega(\text{poly}(n))$)?

   Tasks.

   a) Give an algorithm for COLORED LONGEST PATH running in time $\mathcal{O}^*(2^k)$.

   b) Convince yourself that a LONGEST PATH instance $\mathcal{I} = (G, k)$, it is a YES-instance if and only if there exists a coloring $c: V \to \{1, \ldots k\}$ such that (G, k, c) is a yes-instance of COLORED LONGEST PATH.

   c) Suppose that G has a path P on k vertices $P = (v_1, \ldots, v_k)$. Show that a random k-coloring of $V(G)$ puts each of vertex of P into a different color class with probability at least e^{-k}. This part is the *color coding* technique.

   d) Combining all of the above, find an FPT algorithm for LONGEST PATH with at least constant success rate. What is your running time?

   Hint. You might want to use the following inequalities.

   - $1 + x \leq e^x$ for every $x \in \mathbb{R}$.
   - $k! > (k/e)^k$.
   - **Markov inequality.** For a positively-valued random variable X, $\Pr[X \geq a] \leq \frac{\mathbb{E}[x]}{a}$ for every $a > 0$.

2. In the SET COVER problem, we are given integers n, k, and subsets $\mathcal{S} \subseteq 2^{[n]}$. The goal is to find a subset $\mathcal{T} \subseteq \mathcal{S}$ of size $|\mathcal{T}| \leq k$ such that $\cup_{T \in \mathcal{T}} T \supseteq [n]$. Show that SET COVER is W[2]-hard.

3. In the STEINER TREE problem, we are given an edge-weighted graph G, and a subset $K \subseteq V(G)$. The goal is to find a connected subgraph H of G of minimum weight such that $K \subseteq V(H)$.

   You might have seen on the lecture that if the parameter is $|K|$, then the problem is FPT (with running time $\mathcal{O}^*(3^{|K|})$. Now, we define the parameter to be $h = |V(H)| - |K|$, meaning that the parameter is the amount of vertices of $V(G) \setminus K$ we are allowed to use to build graph H. Show that STEINER TREE with parameter h is W[2]-hard.

   Note. The vertices $V(H) \setminus K$ are called *Steiner vertices* in literature.