- This exam has 4 questions for a total of 150 marks, of which you can score at most 100 marks.
- Use extra sheets to do your rough work. Minimize rough work on the answer sheets.
- You will **not** get the credit for an answer if you do one of the following:
  - Not clearly state all results that you use;
  - Use arguments with implicit assumptions;
  - Use unclear or incomplete reasoning, or appeals to intuition;
  - Use "weasel words" such as "clearly" and "obviously".
- It is OK to cite results that were stated and/or proven in class, and (parts of) existing answers from your own answer sheet. You may also freely use well-known properties of mathematical objects, but you must **clearly state** what it is that you use.
- **You may get partial credit for answers which are partly correct.** So you should consider writing down what you know to be correct, even in those cases where you cannot provide a proof.

1. Recall the MINIMUM SET COVER problem which we discussed in class.

   MINIMUM SET COVER **Parameter:** $n = |\mathcal{U}|$
   Input: A finite universe $\mathcal{U}$ and a collection $\mathcal{S}$ of non-empty subsets of $\mathcal{U}$.
   Question: What is the *minimum size* of a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $(\cup_{X \in \mathcal{S}'} X) = \mathcal{U}$?

   (a) Describe an algorithm which solves MINIMUM SET COVER in FPT time. [20]

   (b) Prove that your algorithm is correct, and that it runs in FPT time. What is the running time bound of your FPT algorithm in the $\mathcal{O}^*()$ notation? [10]

2. For a positive integer t, a *proper* t-colouring of an undirected graph $G = (V, E)$ is any function $c : V(G) \to [t]$ such that $c(x) \neq c(y)$ for each edge $uv \in E$. The *chromatic number* of G is the smallest integer t such that G has a proper colouring with t colours.

   (a) Explain why the problem of finding the chromatic number k of a graph is unlikely to be FPT with k as the parameter. [5]

   (b) State the Principle of Inclusion and Exclusion. [5]

   (c) Describe an algorithm which finds the chromatic number of an input graph $G = (V, E)$ in $\mathcal{O}^*(2^{|V(G)|})$ time. [25]

   (d) Prove that your algorithm is correct, and that it runs in $\mathcal{O}^*(2^{|V(G)|})$ time. [10]

3. In this problem we look at yet another parameterization of the problem of finding the chromatic number of a graph.

   CHROMATIC NUMBER-(q, t) **Parameter:** $q + t$
   Input: Undirected graph G, $q \in \mathbb{N}$, and a tree decomposition $(T, \mathcal{X})$ of G of width t.
   Question: Does G have a proper colouring with at most q colours?

(a) Describe an algorithm which solves CHROMATIC NUMBER-(q, ℓ) in FPT time. [25]

(b) Prove that your algorithm is correct, and that it runs in FPT time. What is the running time bound of your FPT algorithm in the $\mathcal{O}^*()$ notation? [10]

4. In this problem we look at a generalization of proper colouring.

> LIST COLOURING **Parameter:** $n = |V(G)|$
> **Input:** An undirected graph G; for each vertex v of G, a list $L(v) \subseteq \mathbb{N}$ of *admissible* colours of v.
> **Question:** Does G have a *proper* colouring $c : V(G) \to \mathbb{N}$ such that each $v \in V(G)$ satisfies $c(v) \in L(v)$?

(a) Prove that for the sake of finding an FPT algorithm for this problem we may assume, without loss of generality, that the total number of admissible colours (that is, the size of the range of the function L) is at most $2|E(G)|$. [5]

(b) Describe an algorithm which solves LIST COLOURING in FPT time.

- *Hint:* Let t be the total number of admissible colours. Consider the set of all tuples $\langle X_1, X_2, \ldots, X_t \rangle$ where each X_i is either empty, or is a non-empty independent subset of $V(G)$ such that every $x \in X_i$ satisfies $i \in L(x)$. Now apply Inclusion and Exclusion.

[25]

(c) Prove that your algorithm is correct, and that it runs in FPT time. What is the running time bound of your FPT algorithm in the $\mathcal{O}^*()$ notation? [10]