Randomization and derandomization

1. Give a randomized FPT algorithm for the problem of deciding whether a given undirected graph contains a cycle of length at least k. Your algorithm should have running time $c^k n^{O(1)}$. Note that a graph may not have any cycle of length exactly k, but contain a much longer cycle. Derandomize your algorithm using perfect hash families.

2. A graph H is called r-colorable if there exists a function $\chi : V(H) \to [r]$ such that $\chi(u) \neq \chi(v)$ for every $uv \in E(H)$. Consider the following problem: given a perfect graph G and integers k and r, check whether G admits an r-colorable induced subgraph on at least k vertices. Show an algorithm for this problem with running time $f(k; r)n^{O(1)}$. You could use the fact that we can find a maximum independent set in perfect graphs in polynomial time.

3. In the Pseudo Achromatic Number problem, we are given an undirected graph G and a positive integer k, and the goal is to check whether the vertices of G can be partitioned into k groups such that every two groups are connected by an edge. Obtain a randomized algorithm for Pseudo Achromatic Number running in time $2^{O(k^2 \log k)} n^{O(1)}$.

4. Consider a slightly different approach for Subgraph Isomorphism on graphs of bounded degree, where we randomly color vertices (instead of edges) with two colors.

   - Show that this approach leads to $2^{(d+1)k} k! n^{O(1)}$-time Monte Carlo algorithm with false negatives.
   - Improve the dependency on d in running time of the algorithm to $d^{O(k)} k! n^{O(1)}$.

5. Show that, for every $n, \ell \geq 2$, any $(n, 2, \ell)$-splitter needs to contain at least $\log_\ell n$ elements. In other words, show that the $\log n$ dependency in the size bound of splitter is optimal.

6. Give a deterministic version of the first algorithm developed in Exercise 3. Your algorithm should run in time $2^{(d+1)k+o(dk)} k! n^{O(1)}$.