1. (15 pts) **Planar Separators:** Use the planar separator theorem from class to show that there is a fixed constant $c$ such that given any $\epsilon > 0$, one can remove at most $c\epsilon n$ edges to break any planar graph on $n$ vertices into components of size at most $1/\epsilon^2$. Prove this by writing and solving an appropriate recurrence relation.

2. (10 pts) **Faster 100-CNF-Sat:** Solve 100-CNF-Sat in $O^*((2^{\frac{1}{\epsilon}})^n)$ time for some $\epsilon > 0$, where $n$ denotes the number of variables.

3. (15 pts) **Connecting Dots with Lines:** Give an $O^*((k^{4k})$-time algorithm that takes as input $n$ points in the plane $\mathbb{R}^2$ and an integer $k$, and determines whether there exist $k$ straight lines such that every point is on some line. Hint: First, look at $k + 1$ points that are on one line to find a reduction rule. Second, conclude something if $n$ is too large when compared with $k^2$ and your reduction rule does not apply. Third, design an $O^*((n^{2k})$ time algorithm.

4. (10 pts) **Triangle Partition:** A triangle of a graph $G = (V,E)$ is a triple $u,v,w \in V$ such that $(u,v), (v,w), (u,w) \in E$. A triangle partition is a partition of $V$ into triangles, e.g., a set of triangles $T_1, \ldots, T_{n/3}$ such that $T_i \cap T_j = \emptyset$ and $\cup_i T_i = V$.

   • (5pts) Give an algorithm that determines whether there is a triangle partition of a graph on $n$ vertices in $O^*(2^n)$ time. Can you give an algorithm that uses $O^*(2^n)$ time and polynomial space?
   
   • (5pts) Give an algorithm that takes as input a graph $G$ on $n$ vertices and a vertex cover of $G$ of size at most $k$, and determines whether there is a triangle partition of $G$ using $O^*(2^k)$ time. Hint: Use that every vertex not in the vertex cover must be in a different triangle, thus a triangle partition can be formed by letting each vertex not in the vertex cover decide with which vertices it is in a triangle.