Problem 1. Given a graph $G$, its vertex cover $W$, and an integer $k$, the Disjoint Vertex Cover problem asks if there exists a vertex cover of $G$ disjoint from $W$ of size at most $k$. Show a polynomial time algorithm for this problem.

Problem 2. Given a graph $G$ and its vertex cover $Z$ of size $k$, the Vertex Cover Compression problem asks if there exists a vertex cover of $G$ of size $k - 1$. Show an $O^*(2^k)$ time algorithm for this problem. Hint: Call a Disjoint Vertex Cover algorithm $2^k$ times.

Problem 3. Solve Vertex Cover in $O^*(2^k)$ time by calling a $O^*(2^k)$ time algorithm for Vertex Cover Compression $n-k$ times. Hint: Start with a $k$-vertex subgraph of the input graph.

Problem 4. Recall the $2k^2$ kernel for Vertex Cover from the last class. It starts with exhaustively applying two rules: (1) if there is a vertex of degree 0, remove it; (2) if there is a vertex of degree greater than $k$, remove it and decrease $k$ by one. Think of another rule that can let us get rid of vertices of degree 1. What is the size of the improved kernel?

Problem 5. Show a kernel with $O(k^2)$ vertices for Cluster Editing. Hint: use the following three reduction rules.

1. If a vertex is not part of any $P_3$ (induced path on three vertices), delete it.
2. If an edge is a part of more than $k$ $P_3$’s, delete it and decrease $k$ by one.
3. If a non-edge is a part of more than $k$ $P_3$’s, add it and decrease $k$ by one.

Problem 6. Given a 2-CNF formula and an integer $k$, the Min-2-CNF-SET problem asks to find an assignment which satisfies at most $k$ clauses. Show an $O^*(2^k)$ time algorithm for this problem.

Problem 7. In the class we defined a problem to be FPT if it admits an $f(k) \cdot n^c$ time algorithm. Show that if we changed the bound to $f(k) + n^c$ we would get an equivalent definition.

Problem 8. Show that an $O((\log n)^k \cdot n)$ time algorithm is an FPT algorithm.

This problem set adds 3 points to the threshold for grade 4.0, and 6 points for 6.0