Problem 1

Let $G = (V, E)$ be a directed graph, and let $k$ be a positive integer. We call a set $E' \subseteq E$ a cycle cut if removing the edges of $E'$ from $G$ results in a graph without directed cycles. We consider the cycle cut to be of size $k$ if $|E'| = k$. Now consider the following language:

$\text{CYCLE-CUT} = \{ \langle G, k \rangle \mid G \text{ has a cycle cut of size } \leq k \}$

Prove that $\text{CYCLE-CUT}$ is NP-Complete via reduction from vertex cover.

Solution:

$\text{VERTEX-COVER} = \{ \langle G, k \rangle \mid G \text{ is an undirected graph that has a cover of size } \leq k \}$

(aka every edge of $G$ is adjacent to one of the vertices in the cover)

Part 1: $\text{CYCLE-CUT}$ is in NP aka a candidate cycle cut can be verified in polynomial time

The verifier for $\text{CYCLE-CUT}$ is as follows.

On input $\langle G, k, c \rangle$:

1. Check that $E'$ includes $k$ valid edges of $E$.
2. Generate the graph $G'' = (V, E - E')$.
3. Perform a depth first search on $G''$, marking nodes that you’ve visited. If at any point you return to a previously visited node, reject.
4. If checks 1 and 3 pass, accept; otherwise, reject.

Since this verification procedure is comprised only of polynomial-time subroutines, the overall verification procedure will also run in polynomial time.

Part 2: $\text{CYCLE-CUT}$ is NP-HARD via reduction from vertex cover. The reduction will work as follows:

1. For every vertex $V$ in $G$, add an additional vertex $V'$ with a directed edge $V \rightarrow V'$.
2. For every undirected edge $E = (U, V)$ in $G$, replace it with two directed edges $E' = V' \rightarrow U$ and $E'' = U' \rightarrow V$.
3. Output this new graph $G'$.

This reduction requires two loops, one through the vertices of $G$ and one through the edges. Since both of these loops are linear with respect to the input, this reduction will run in polynomial time.

If $G$ has a vertex cover of size $\leq k$, then $G'$ has a cycle cut of size $\leq k$. Our new graph, $G'$ contains a directed cycle for every edge in $G$, as well as potentially any cycle that was in the original $G$ (depending on which way you drew the directed edges). However, the directed edge
between each vertex pair $V \rightarrow V'$ acts a switch. Any cycle passing through $V$ can only remain intact if the edge $V \rightarrow V'$ remains in the graph. A vertex cover of size $k$ implies that every edge in the graph is connected to one of these $k$ vertices at one end. For these $k$ vertices in $G$ then, removing the corresponding $V \rightarrow V'$ edge in $G'$ will eliminate any possible cycles.

**If $G$ does not have a vertex cover of size $\leq k$, then $G'$ does not have a cycle cut of size $\leq k$.** If $G$ does not have a vertex cover of size $k$, that means that at least one edge will not have a member of the cover at either end. Based on the construction of $G'$ then, there will be at least one cycle that cannot be removed.

Thus CYCLE-CUT is in NP and is NP-HARD, thus it is NP-Complete.
Problem 2

“A little bit of everything” 3SAT is the problem of determining whether a boolean formula in conjunctive normal form (having exactly 3 literals in each clause) is satisfiable such that at least one clause has all three literals evaluate to true, at least one clause has exactly two literals evaluate to true, and at least one clause has exactly one literal true.

Prove that “A little bit of everything” 3SAT (LBE) is NP-complete, do a reduction from 3SAT.

Solution:

Part 1: LBE is in NP aka a candidate assignment can be verified in polynomial time

The verifier for LBE is as follows.
On input $\langle \phi, c \rangle$:

1. Check that $c$ is satisfiable.
2. Check that there is a clause with three true literals.
3. Check that there is a clause with exactly two true literals.
4. Check that there is a clause with exactly one true literals.
5. If checks 1 - 3 pass, accept; otherwise, reject.

Since this verification procedure is comprised only of $O(n)$ time checks, the overall verification procedure will also run in polynomial time.

Part 2: LBE is NP-HARD via reduction from 3SAT. Define the following function:

$$f(\phi) = \phi \land (x \lor y \lor z) \land (x \lor y \lor \overline{z}) \land (x \lor \overline{y} \lor \overline{z})$$

Where $x, y,$ and $z$ are variables that do not appear in $\phi$.

In linear $O(n)$ time we can find three new variables to use and append the defined clauses onto $\phi$. Thus this reduction will take polynomial time.

If $\phi \in 3\text{SAT}$, then $f(\phi) \in \text{LBE}$. Since $x, y,$ and $z$ are not in $\phi$, they can be assigned independently of the assignment that satisfies $\phi$. We can assign $x, y,$ and $z$ to true, which will make our first appended clause all true, the second appended clause will have exactly two true literals, and the third will have exactly one true literal. Thus $f(\phi) \in \text{LBE}$.

If $f(\phi) \in \text{LBE}$, then $\phi \in 3\text{SAT}$. If $f(\phi) \in \text{LBE}$, then we know that it has a satisfying assignment with the prescribed makeup. This immediately implies that $\phi$ has a satisfying assignment since the extra clauses in $f(\phi)$ don’t not share any variables with $\phi$. 