Professor Toole proposes a new divide-and-conquer algorithm for computing minimum spanning trees, which goes as follows. Given a graph $G = (V, E)$, partition the set of vertices into two sets $V_1$ and $V_2$ such that $|V_1|$ and $|V_2|$ differ by at most 1. Let $E_1$ be the set of edges that are incident only on vertices in $V_1$, and let $E_2$ be the set of edges that are incident only on vertices in $V_2$. Recursively solve a minimum-spanning-tree problem on each of the two subgraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Finally, select the minimum-weight edge in $E$ that crosses the cut $(V_1, V_2)$, and use this edge to unite the resulting two minimum spanning trees into a single spanning tree.

Either argue that the algorithm correctly computes a minimum spanning tree of $G$, or provide an example for which the algorithm fails.

We argue that the algorithm fails. Consider the graph $G$ below. We partition $G$ into $V_1$ and $V_2$ as follows: $V_1 = \{A, B\}$, $V_2 = \{C, D\}$. $E_1 = \{(A, B)\}$. $E_2 = \{(C, D)\}$. The set of edges that cross the cut is $E_c = \{(A, C), (B, D)\}$.

![Graph Diagram]

Now, we must recursively find the minimum spanning trees of $G_1$ and $G_2$. We can see that in this case, $\text{MST}(G_1) = G_1$ and $\text{MST}(G_2) = G_2$. The minimum spanning trees of $G_1$ and $G_2$ are shown below on the left.

The minimum weighted edge of the two edges across the cut is edge $(A, C)$. So $(A, C)$ is used to connect $G_1$ and $G_2$. This is the minimum spanning tree returned by Professor O’Toole’s algorithm. It is shown below and to the right.

![Minimum Spanning Tree Diagram]

We can see that the minimum-spanning tree returned by Professor O’Toole’s algorithm is not the minimum spanning tree of $G$, therefore, his algorithm fails.