A ADDITIONAL PROOF SKETCHES

In this section, we provide additional explanations and proof sketches for the correctness of the oracles in Table 1.

A.1 Strongly Connected Components

The task of Strongly Connected Components is defined as follows.

Problem A.1 (Strongly Connected Components [8]). *Given a directed graph G, return* $\texttt{res}(G)$ *as the set of all Strongly Connected Components (SCCs) from G, where SCC is a maximal subset of vertices where each vertex in the subset is reachable from every other vertex.*

Lossless-Cutting. Given a graph G and its dividing subgraphs $G^{(1)}$ and $G^{(2)}$, the Lossless-Cutting oracle shown in Table 1 states that, $\texttt{res}(G) = \texttt{res}(G^{(1)}) \cup \texttt{res}(G^{(2)})$ if there is no cutting edge in G connecting two vertices belonging to $G^{(1)}$ and $G^{(2)}$, respectively.

Proof sketch. We first show that, for each SCC $H \in \texttt{res}(G)$, either $H \in \texttt{res}(G^{(1)})$ or $H \in \texttt{res}(G^{(2)})$. Since there is no cutting edge, H cannot simultaneously contain vertices from both $G^{(1)}$ and $G^{(2)}$, as this would prevent the vertices belonging to different subgraphs from reaching each other. Hence, H is either a subgraph of $G^{(1)}$ or $G^{(2)}$. In addition, H is still maximal in $G^{(1)}$ and $G^{(2)}$ as they are subgraphs of G, which means that H is either in $\texttt{res}(G^{(1)})$ or $\texttt{res}(G^{(2)})$. Similarly, we can prove the converse, *i.e.,* for each $H \in \texttt{res}(G^{(1)})$ or $H \in \texttt{res}(G^{(2)})$, $H \in \texttt{res}(G)$. Therefore, we have $\texttt{res}(G) = \texttt{res}(G^{(1)}) \cup \texttt{res}(G^{(2)})$. □

Lossy-Cutting. The Lossy-Cutting oracle shown in Table 1 for SCC states that, after removing all SCC containing cutting edges from $\texttt{res}(G)$, it will become a subset of $\texttt{res}(G^{(1)}) \cup \texttt{res}(G^{(2)})$.

Proof sketch. Note that for each SCC H in $\texttt{res}(G)$ that does not contain any cutting edges, it can only include vertices from one side of $G^{(1)}$ or $G^{(2)}$. Therefore, we only need to prove that those SCCs are either in $\texttt{res}(G^{(1)})$ or $\texttt{res}(G^{(2)})$, which we have proven in the proof of the correctness of Lossless-Cutting for SCC. □

A.2 All Pairs Shortest Path

The task of All Pairs Shortest Path is defined as follows.

Problem A.2 (All Pairs Shortest Path (APSP) [9]). *Given a weighted, undirected graph G, return a 2D table* $\texttt{res}(G)$ *such that for each pair of vertices* $(u, v) \in G$, $\texttt{res}(G)[u][v]$ *represents the length of the shortest path between u and v.*

Given a graph G and its dividing subgraphs $G^{(1)}$ and $G^{(2)}$, the Lossy-Cutting for APSP, as shown in Table 1, states that for every pair of vertices (u, v) that are either both in $G^{(1)}$ or both in $G^{(2)}$, the length of the shortest path between them in the respective subgraph is no less than the length of which in the original graph. *i.e.,* either $\texttt{res}(G^{(1)})[u][v] \geq \texttt{res}(G)[u][v]$ or $\texttt{res}(G^{(2)})[u][v] \geq \texttt{res}(G)[u][v]$.

Proof sketch. Since all paths between any pair of vertices u and v in the dividing subgraphs will remain in the original graph, it is easy to see that the correctness of the oracle holds. □

A.3 Minimum Spanning Tree

The task of Minimum Spanning Tree is defined as follows.

Problem A.3 (Minimum Spanning Tree (MST) [10]). *Given an undirected and positive weighted graph G, return* $\texttt{res}(G)$ *as a set of edges in a minimum spanning tree or forest on G.*

Given a graph G and its dividing subgraphs $G^{(1)}$ and $G^{(2)}$, the Lossy-Cutting for MST, as shown in Table 1, states that, after excluding all cutting edges in $\texttt{res}(G)$, the weight of the minimum spanning tree or forest on G is no greater than the sum of weights on subgraphs.

Proof sketch. After excluding all cutting edges in $\texttt{res}(G)$, we only have two types of edges in the MST, connecting vertices that are either both in $G^{(1)}$ or both in $G^{(2)}$. We denote the two types of edges as $E^{(1)}$ and $E^{(2)}$, respectively. After that, we need to prove that, the sum of edge weights over $E^{(1)}$ and $E^{(2)}$ is no greater than which over $\texttt{res}(G^{(1)})$ and $\texttt{res}(G^{(2)})$. Actually, we can show that

$$\sum_{(u,v,w) \in E^{(1)}} w \leq \sum_{(u,v,w) \in \texttt{res}(G^{(1)})} w,$$
$$\sum_{(u,v,w) \in E^{(2)}} w \leq \sum_{(u,v,w) \in \texttt{res}(G^{(2)})} w,$$

where w represents the edge weight.

First, if $E^{(1)}$ is a spanning tree of $G^{(1)}$, its weight cannot be greater than the weight of $\texttt{res}(G^{(1)})$; otherwise, we could replace $E^{(1)}$ with $\texttt{res}(G^{(1)})$ in $\texttt{res}(G)$ to obtain a spanning tree of G with a smaller weight.

On the other hand, if $E^{(1)}$ is not a spanning tree of $G^{(1)}$, we can still replace $E^{(1)}$ with $\texttt{res}(G^{(1)})$ in $\texttt{res}(G)$. Although this will no longer form a spanning tree due to the presence of redundant edges, we can repeatedly remove one edge from each cycle until the remaining set of edges forms a spanning tree. Since all edge weights are positive, this process can only reduce the weight, ultimately resulting in a spanning tree of G with a smaller weight.

The proof for $E^{(2)}$ and $\texttt{res}(G^{(2)})$ is similar, and we ultimately prove the correctness of the oracle by summing the two inequalities. □

[8]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.components.strongly_connected_components.html

[9]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.shortest_paths.unweighted.all_pairs_shortest_path.html

[10]https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.tree.mst.minimum_spanning_tree.html