Let $G = (V, E, w)$ be a directed and weighted graph with positive edge weights $w(e) > 0$ for each edge $e \in E$. For a pair of vertices $u, v \in V$, there may be *multiple* shortest paths from u to v. Let $\Pi_{u,v}$ denote all such paths. In other words, for a pair of vertices u and v, $\Pi_{u,v}$ is the set of all shortest paths from u to v.

A vertex x is called *useful* if x lies on *any* path in $\Pi_{u,v}$. Given the graph G and a pair of vertices $u, v \in V$, describe an $O(m \log n)$ algorithm to return all useful vertices.

Solution. To find all useful vertices in G defined on path $\Pi_{u,v}$, we first run Dijkstra's algorithm from u to get shortest path distances. We must now repeat this for vertex v, however we must create a new graph with all edges reversed in order to be able to run Dijkstra's algorithm from this vertex. Then, for each vertex $x \in V$, if the distance to x from u plus the distance distance from v to x (in the reversed graph) equals the shortest distance between u and v, then the point is useful. Append each of these vertices to a list and return it. This approach has a time complexity of $O(m \log n + n)$ as Dijkstra's algorithm is in $O(m \log n)$ time, and we may visit all vertices n.