Dynamic programming

Kruskal's algorithm to find a minimum spanning tree of a weighted connected graph G = (V, E).

- Start with a forest of 0 trees, where each vertex is a separate tree.
- Find the edge with the minimum weight that connects two trees.
- Merge the two trees into a single tree.

Algorithm:
1. Sort all edges in non-decreasing order of weight.
2. Initialize a set S containing the forests.
3. For each edge (u, v) sorted in increasing order of weight:
   - If u and v are in different trees, merge the trees.
   - Add the edge to the solution set T.
4. Return T as the minimum spanning tree.

Analysis:
- Time complexity: O(E log E) using a priority queue.
- Space complexity: O(V + E) for the set of forests and the edges.

Example:
Given a graph G with vertices V and edges E, find the minimum spanning tree T.

Proof of correctness:
- By the cut property, an edge not in the minimum spanning tree can never be added.
- By the cycle property, an edge added to the tree must not form a cycle.

Note: The diagram illustrates the process of merging trees as edges with decreasing weights are added to the solution set.