

Exercise 4.3.32 Specified Set: Given a **connected edge-weighted** graph G and a specified **set of edges S** (having no cycles), describe a way to **find a minimum-weight spanning tree** of G that **contains all the edges in S** .

Solution: We will use a modification of Kruskals algorithm to compute the MST

- Step 1: Sort edges of G by weight in ascending order
- Step 2: Iterate through the sorted edges
- Step 2.1: For each edge seen in the iterator, if it is not contained in the set of edges S , then skip. Remember, we want to add the edges that are only seen in the set of edges S
- Step 2.2: If it is contained in S
- Step 2.2.1: AND if it GENERATES a cycle, then do NOT add this edge to the MST. So just skip
- Step 2.2.2: AND if it DOES NOT GENERATE a cycle, add it to the MST.
- Step 3: Finish iteration for all edges processed.
- Step 4: Return MST

Additionally, we can verify if valid MST using DFS.