Algorithm 1 Kruskal Minimum Spanning Tree Algorithm

$WeightedGraph(G)$
▷ let P be a partition of the vertices of G, where each vertex forms a separate set ▷ let Q be a priority queue storing the edges of G and their weights
$T \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
  $(u, v) \leftarrow Q.removeMinElement$
  if $P.find(u) \neq P.find(v)$ **then** ▷ Ensure u and v are not on same tree
    $T \leftarrow edge(u, v)$ ▷ Assign edge (u,v) to minimum spanning tree
    $P.union(u, v)$ ▷ Replace seperate sets storing u and v with union
  end if
end while
  return $MinimumSpanningTreeT$