1. **Generic MST Algorithm:**

   **Note:** $G(V, E)$ is a given connected undirected graph and $w$ is a function that assigns a weight (real number) to each edge. At the end of the algorithm, the set $A \subseteq E$ contains the edges in an MST. (The notion of a “safe edge” will be discussed in class.)

   **Generic-MST** $(G, w)$

   1. $A = \emptyset$.

   2. while (A does not form a spanning tree) do

      (a) Find a safe edge $\{u, v\}$.

      (b) Add $\{u, v\}$ to $A$.

   3. return $A$.

2. **Kruskal’s MST Algorithm:**

   **Note:** The following outline uses UNION and FIND operations on disjoint sets. These will be defined in the lecture.

   **MST-Kruskal** $(G, w)$

   1. $A = \emptyset$.

   2. for each vertex $v \in V$ do

      MAKE-SET($v$).

   3. Sort the edges of $E$ by non-decreasing weight.

   4. for each edge $\{u, v\} \in E$ in order by non-decreasing weight do

      if (FIND-SET($u$) $\neq$ FIND-SET($v$)) then

         (a) Add $\{u, v\}$ to $A$.

         (b) UNION ($u, v$).

   5. return $A$. (over)
3. Prim’s MST Algorithm:

Notes:

- We assume that a root $r$ of an MST is also given.
- The implementation uses a priority queue (heap) $Q$ which contains a key value for each node that is not currently in the tree.
- For each node $v$ that is not currently in the tree, $\text{key}[v]$ is the minimum weight among the edges that join $v$ to some node in the tree. In other words, $\text{key}[v]$ is the cheapest cost of adding $v$ to the tree.
- $\pi[v]$ is the node $u$ such that $w(v, \pi[v]) = \text{key}[v]$. (Think of $\pi[v]$ as the parent of $v$ in the tree.)
- The algorithm terminates when $Q$ becomes empty.
- The edge set $A$ of the MST computed by the algorithm is given by $A = \{ \{v, \pi[v]\} : v \in V - \{r\} \}$.

MST-Prim $(G, w, r)$

1. for each vertex $v \in V$ do
   
   Set $\text{key}[v] = \infty$ and $\pi[v] = \text{NULL}$.

2. $\text{key}[r] = 0$.

3. Create a priority queue $Q$ containing all the nodes in $V$.

4. while $(Q$ is not empty$)$ do
   
   (a) $u = \text{Extract-Min}(Q)$.

   (b) for each $v \in \text{Adj}[u]$ do
       
       if $((v \in Q) \text{ and } (w(u, v) < \text{key}[v]))$ then
       
       (i) $\pi[v] = u$.
       
       (ii) $\text{key}[v] = w(u, v)$.

   enddo

endwhile