9 Kruskal's Algorithm

Algorithm 25 KRUSKAL'S ALGORITHM

Input: A connected undirected graph $G = (V, E)$
Output: Minimum weight spanning tree T of G

1: $n \leftarrow |G.V|$ $\quad$ ▷ Number of vertices in G
2: $m \leftarrow |G.E|$ $\quad$ ▷ Number of edges in G
3: $T \leftarrow \emptyset$ $\quad$ ▷ Initialize minimum weight spanning tree
$\Big\}\ O(1)$
4: Sort the edges of $G.E$ in non-decreasing order of their weight — $O(m \log m)$
5: **for each** vertex $u \in G.V$ **do**
6: $\quad$ MAKE-SET(u) $\quad$ ▷ Create a separate component for each vertex
7: **end for**
$\Big\}\ O(n)$
8: **for each** edge $(u, v) \in G.E$ taken in non-decreasing order of their weight **do**
9: $\quad$ $\text{Rep}_u \leftarrow$ FIND-SET(u) $\quad$ ▷ Find the component where u belongs
10: $\quad$ $\text{Rep}_v \leftarrow$ FIND-SET(v) $\quad$ ▷ Find the component where v belongs
11: $\quad$ **if** $\text{Rep}_u \neq \text{Rep}_v$ **then** $\quad$ ▷ u and v are from different component
12: $\quad\quad$ $T \leftarrow T \cup \{(u, v)\}$ $\quad$ ▷ Add edge (u, v) to the minimum weight spanning tree T
13: $\quad\quad$ UNION$(\text{Rep}_u, \text{Rep}_v)$ $\quad$ ▷ Connect u and v
14: $\quad$ **else** $\quad$ ▷ u and v are from the same component
15: $\quad\quad$ Do nothing $\quad$ ▷ Connecting u and v will give a cycle
16: $\quad$ **end if**
17: **end for**
18: **return** T

MAKE-SET()	FIND-SET()	UNION()
$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$

Table 1: Time complexity using tree representation for disjoint set data-structure.

Loop Analysis

* For each edge (u, v) we are calling FIND-SET 2 time. (See line 9 and 10)
* Total (max.) call to FIND-SET $= 2 * m$
* Time by all FIND-SET $= (2m)(\log n) = O(m \log n)$
* Total call to UNION $= n-1$ as MST has $n-1$ edges.
* Time by all UNION $= (n-1)(O(1)) = O(n)$

$$T = \underbrace{O(1)}_{1-3} + \underbrace{O(m \log m)}_{4} + \underbrace{O(n)}_{5-7} + \underbrace{O(m \log n)}_{\text{FINDSET}} + \underbrace{O(n)}_{\text{UNION}}$$

$$= \underline{O(m \log m)}$$