CS3000: Algorithms & Data — Summer '25 — Laney Strange

Kruskal's Algorithm

Kruskal's algorithm finds a Minimum Spanning Tree for a given graph. it takes in a graph G and a weight function w. It creates and returns a set A of edges to represent an MST. It is a greedy algorithm that repeatedly finds a "safe" edge with the lowest weight. (Note that a safe edge is an edge (u, v) that we can add to A because $A \cup \{(u, v)\}$ is a subset of an MST.)

Kruskal's calls a few set procedures that we can assume exist and work as intended: Make-Set to create a set with a given element, and Find-Set to determine the set an element belongs to. It also calls Union to union two trees together.

Kruskal(G, w)

```
A = {}
for each vertex v ∈ G.V
    Make-Set(v)
create a single list of the edges in G.E
sort the list of edges into monotonically increasing order by weight w
for each edge (u, v) taken from the list in sorted order
    if Find-Set(u) ≠ Find-Set(v)
        A = A ∪ {(u, v)}
        Union(u, v)
return A
```

We typeset the Kruskal procedure above with the following LaTeX:

```
\begin{codebox}
\Procname{$\proc{Kruskal}(G, w)$}
\li $A = \{\}$
\li \For each vertex $v \in G.V$
\Do
\li $\proc{Make-Set}(v)$
\End
\li create a single list of the edges in $G.E$
\li sort the list of edges into monotonically increasing order by weight $w$
\li \For each edge $(u, v)$ taken from the list in sorted order
\Do
\li \If $\proc{Find-Set}(u) \ne \proc{Find-Set}(v)$
\Then
\li $A = A \cup \{(u, v)\}$
\li $\proc{Union}(u, v)$
\End
\End
\li \Return $A$
\end{codebox}
```