20 - Spanning Trees

Trees
Undirected Graphs
Rooted Tree vs.
Free Tree

MST - Spanning Tree with least weight
To find a minimum spanning tree:
Start with an empty set of edges - $S$
While size of $S < |V| - 1$ {
    add an edge from the MST
}

Cut of a graph is a partition of the vertices

$S =$ set of purple edges
Kruskal's Algorithm

\[ E = O(v^2) \]
\[ O(E \log E) \]
\[ = O(E \log V^2) = O(2E \log V) \]
\[ O(E \log V) \]

1. Sort edges \( O(E \log E) \)
2. While size \( S < |V| - 1 \) 
   1. Add edge to set
   2. Merge 2 sets
   3. Check to see if cycle is created
\[ O((V+E) \log^* V) \]

\( S = \emptyset \ \ O(1) \)
Prim's Algorithm

$S = \emptyset \quad O(1)$

Start with a vertex $O(1)$

While size $S < |V| - 1$ and add lightest edge that doesn't create a cycle that's connect.

to your current tree

Priority Queue
- Add all edges to PQ \( O(E) \)
  - Those not connecting to start have \( \infty \) weight

Inside
- Extract-Min \( O(\lg E) \) \( \quad \text{V-1} \)
- For every edge leaving new node \( E \)
  - Decrease-Key \( O(\lg E) \)
  - For every edge leaving new node that makes cycle \( \text{Delete} \quad O(\lg E) \)

\[ \text{WC: } O(E \lg E + V \lg E) = O(E \lg V) \]
\[ \text{AM: } O(E + V \lg E) \text{ w/ Fb Heap} \]