May 19

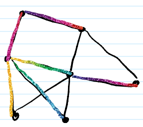
Wednesday, May 19, 2021 12:09 PM

Comp Sci 212

1. Minimum Spanning Trees

2. Kruskal's Algorithm

Def. Spanning tree of a graph $G = (V, E)$ is a subgraph $G' = (V, E')$ $(E' \subseteq E)$, such that G' is a tree. $(|E'| = |V| - 1)$



- Minimal subgraph containing all vertices but still connected

Weighted graphs

$G = (V, E, w)$, $w: E \to \mathbb{R}$

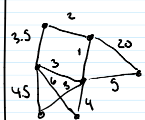
Transportation networks

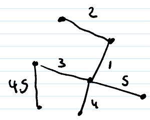
Internet networks

Social networks

weights represent costs (non-negative)

Problem - How to find the spanning tree where the sum of the weights is minimal? (Minimum Spanning Tree)





Graphs can have multiple MST's, (if all edges are the same weight)

Kruskal's algorithm - G is connected

1. Sort the edges
2. From the smallest edge to the largest edge
3. For each edge, add it if it doesn't create a cycle.

Theorem. Kruskal's algorithm outputs a spanning tree (T).

Proof. T is acyclic. Need to prove that T is connected. Use contradiction, assume T is not connected. Let u, w be disconnected in T. u, w are connected in original graph G $(v_0, v_1, v_2, \dots, v_n, v_{n+1})$. There is some i s.t. v_i, v_{i+1} not connected in T.
$u = v_0$, $w = v_{n+1}$

Kruskal's algorithm considers the edge $\{v_i, v_{i+1}\}$, but didn't add it. Thus, adding $\{v_i, v_{i+1}\}$ would create a cycle. This means that v_i, v_{i+1} are connected, which is a contradiction. $\square$

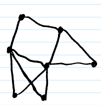
Theorem. Kruskal's algorithm outputs a minimum spanning tree.

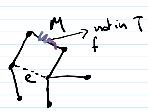
(every edge in G has a unique weight)

Proof. Let T be tree output by Kruskal's algorithm.

Assume that T is not an MST. Let M be an MST, let e be the edge with least weight in T, but not in M. (Every edge in T with smaller weight is in M).

Adding e to M creates a cycle. This cycle must have an edge f not in T. $M' = (M \setminus \{f\}) \cup \{e\}$. M' is also a spanning tree. (should be a spanning tree).





We will show that M' has smaller weight than M.

weight of M' = weight of $M - w(f) + w(e)$

Either, $w(f) > w(e)$ or $w(e) > w(f)$
Case 1 Case 2

Case 1 - weight of M' < weight of M

Case 2 - $w(f) < w(e)$. Kruskal's algorithm did not add f, so adding f would have created a cycle. All edges in this cycle (besides f), are in T, and have a smaller weight than e, thus, they're also in M. f is also in M. So M contains a cycle, contradicting the fact that M is a tree. $\square$

Case 3 - $w(f) = w(e)$ - repeat for M'.