KRUSKALS MINIMAL SPANNING TREE

Graph: vertices and edges

cyclical: it's possible to get back to an origin through other vertices

undirected: no arrows

complete: Every vertex is connected to every other vertex

a loop: both endpoints are the same vertex

complete graph on 5 vertices can be shorthanded to K4. This is the most dense graph and the most dense simple graph

a simple graph - no loops or double edges

tree: connected acyclical graph

tree: number of edges is one less than the number of vertices → e = v-1 and v = e + 1

A minimal spanning tree tries to find the lowest "cost" path to connect all vertices of a simple weighted graph.

Algorithm: kruskals minimum spanning tree algorithm

Imagine a graph where each edge has a weight.

Minimal Spanning TREE Array: Index 1: Shortest, Index 2: second shortest etc.

1. sort the edges by weight from smallest to largest

   A→ B Cost: 1, B→C Cost: 3, etc etc → 6,8,10

2. Be greedy: Add each next edge with the least weight, but never add an edge that creates a cycle.
3. We're done when there is one less edge than vertices

To Detect CYCLES:

ones represent that B and C are a part of tree 1. 2s represent that E and F are a part of tree 2

-1 represents no connection.

Index	0	1	2	3	4
Weight	1	2	4	5	5
Vertex 1	B	E	A	B	C
Vertex 2	C	F	B	E	D

Undirected Connections.	A	B	C	D	E	F	
None	-1	-1	-1	-1	-1	-1	
B< - > C	-1	1	1	-1	-1	-1	Add B→C from index 0 since it's cheapest. Both values become a 1.
E < - > F	-1	1	1	-1	2	2	Add E→F as its own tree. We know this doesn;t connect with tree 1 because neither vertex has a 1 in it. Represent new tree with a 2.
Add A < - > B	1	1	1	-1	2	2	Add A→B. One of these is a 1 and the other is a -1, so we are joining a tree.
Add B < - >E	1	1	1	-1	1	1	Add B→E. This connects the two trees together, so convert 2s to 1s.
Add C < - >D	1	1	1	1	1	1	Add C→E. Connects the last point to the tree.

Efficiency: O(V^2 for the connections + E log E for sorting the edges)

[v * (v-1)]/2 → the number of edges in worst case is v^2 in a complete graph. → O(v^2 + V^2 * log(V^2)) = O(V^2 + V^2 * logV) → polynomial work

PRIMS algorithm is next: far better if things are very very dense.