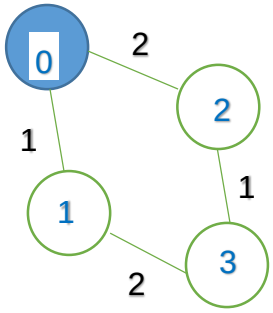
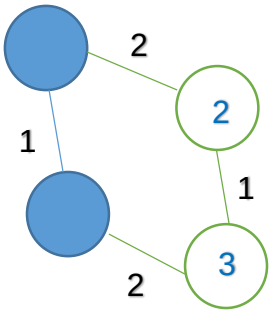
Task

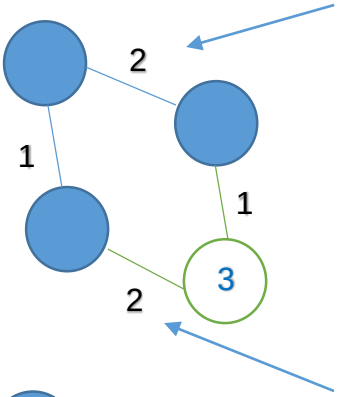
- Give graph
- Explain how MST generated
- Explain differences

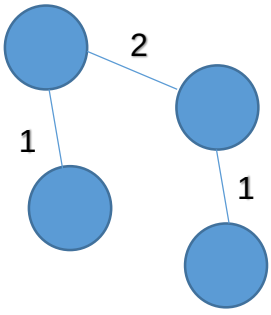
Prim's



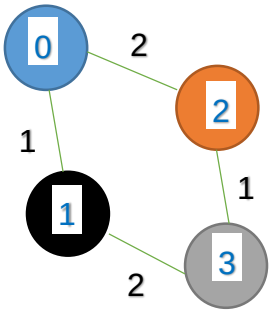


Same weight however this edge was added to the queue first

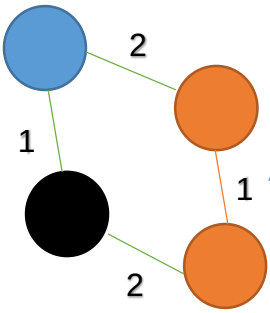


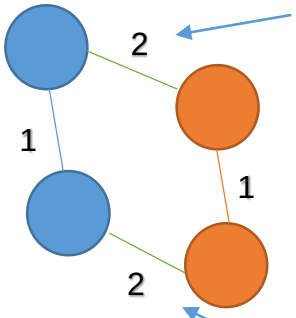


Kruskal's

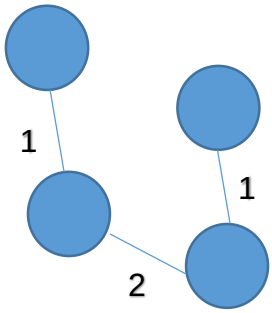


Same weight so it depends on who entered the queue first, could be random





Same weight so it depends on who entered the queue first, could be random



Prim's and Kruskal both return the minimum spanning tree (MST) of a graph, however because a graph can have multiple MST's when weights are equal, the way the code is run for the two different MST algorithms can result in different MST's while still both being correct.

Kruskal's algorithm focuses on avoiding cycles and uses a tree format to do so. Krustal's also begins by having a "global" view of all the edges. The way these are sorted when equal weights appear are random as opposed to Prim's. This is where the chance of divergence arises. Prim's algorithm methodically starts at node 0 and branches out by populating a priorityqueue with edges as it encounters them via vertices that the algorithm has visited. While Krustal's also uses a priorityqueue, they way the edges are added leaves room for a divergence from the exact same MST every time. This is how it's possible for Prim's and Kruskal's to result in different MST's for the same graph input.