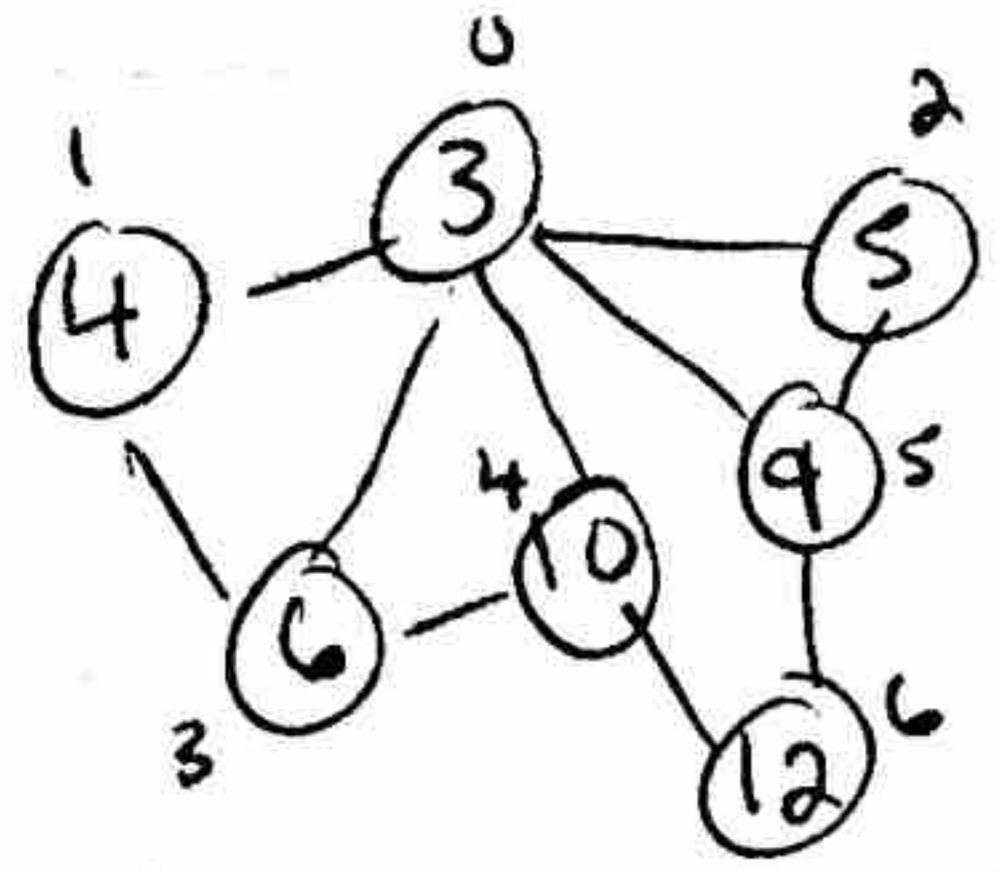


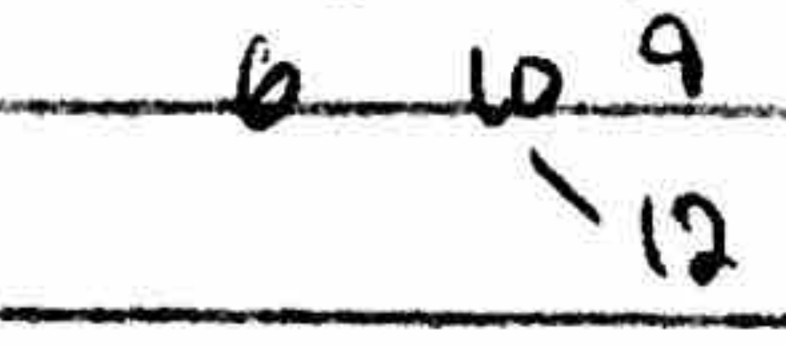
- All weights are k, where k > Ø.
- the number by each vertex is the index in the List<Edg>[] in the graph

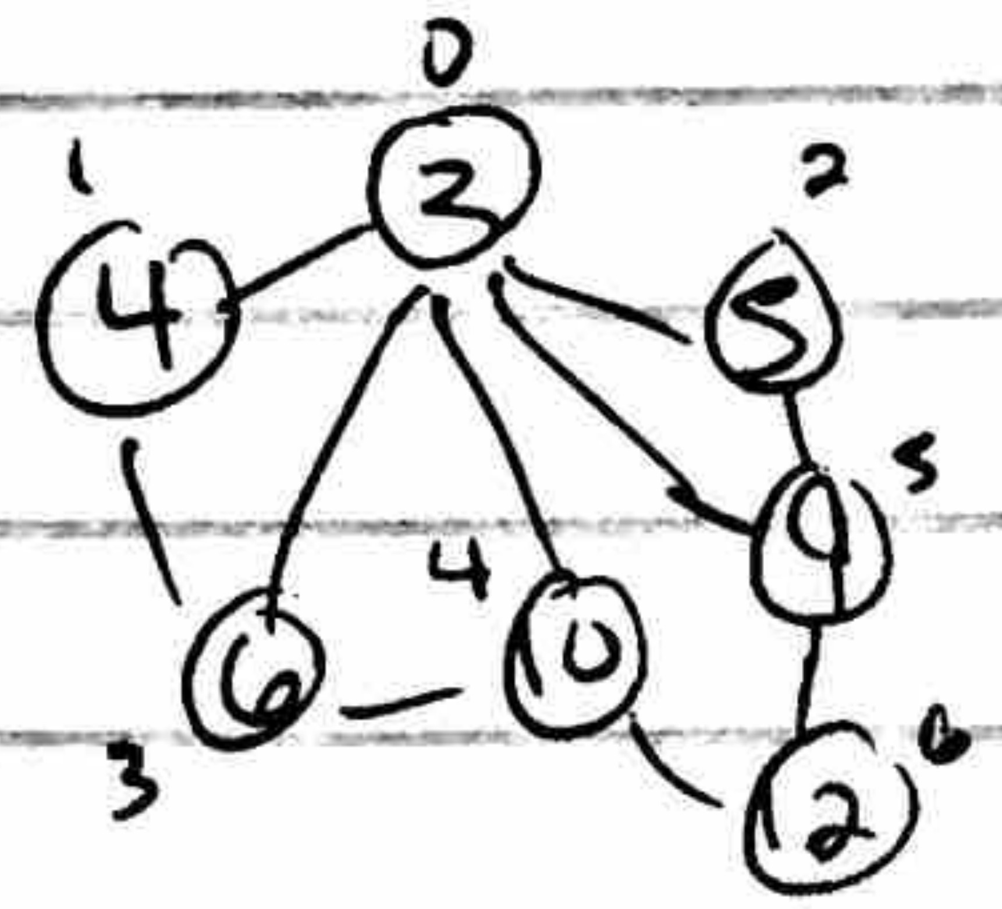
Prim's Algorithm

- starts with index Ø and add all incoming edges to queue (PQ)
  queue = 4, 5, 6, 10, 9 ; visited = Ø
- then we get the highest priority element in PQ, since they all have same weight/priority (k), the are removed like normal queue (FIFO)
  - edge is 4->3
  - since 1 (index of 4) is not in visited, we add this edge to MST, then add 1 to visited and incoming nodes to the queue
  - queue = 5, 6, 10, 9 ; visited = Ø, 1

- MST currently looks like

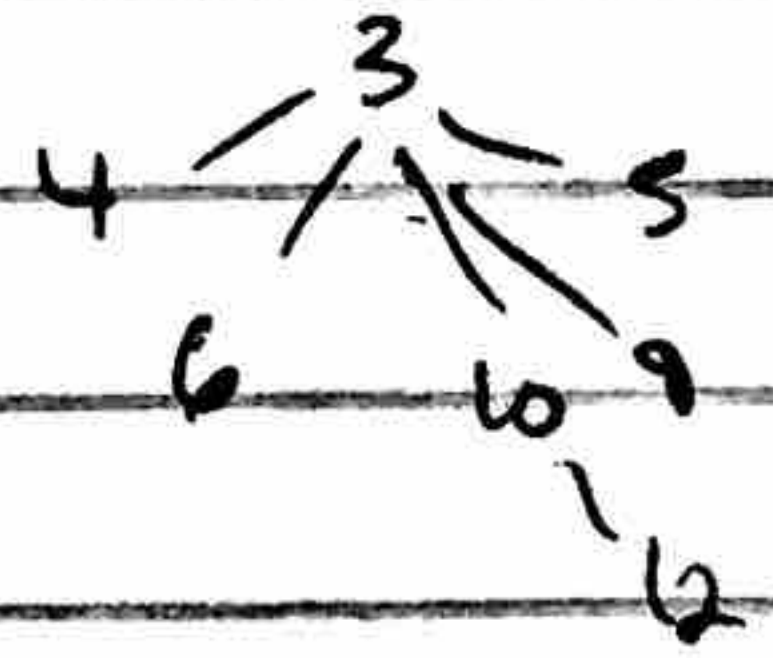
- after Prim's algorithm is complete, the MST would look like this:





- Kruskal's Algorithm
  - would add all edges to PQ

    queue = 4->3, 5->3, 6->3, 10->3, 9->3, 3->4, 6->4, 3->5, 9->5, 3->6, 4->6,
    (source->target) 10->6, 3->10, 6->10, 12->10, 3->9, 5->9, 12->9, 10->12, 9->12.
  - it would poll each edge, and if it didn't already exist in tree, it would add it to MST.



- In this case, both Prim's and Kruskals created the same MST, but if the same graph had been created, with the only difference being the order in which the edges and vertrces were added (between Prim's and Krsuskal's), then the two would likely generate different MSTs