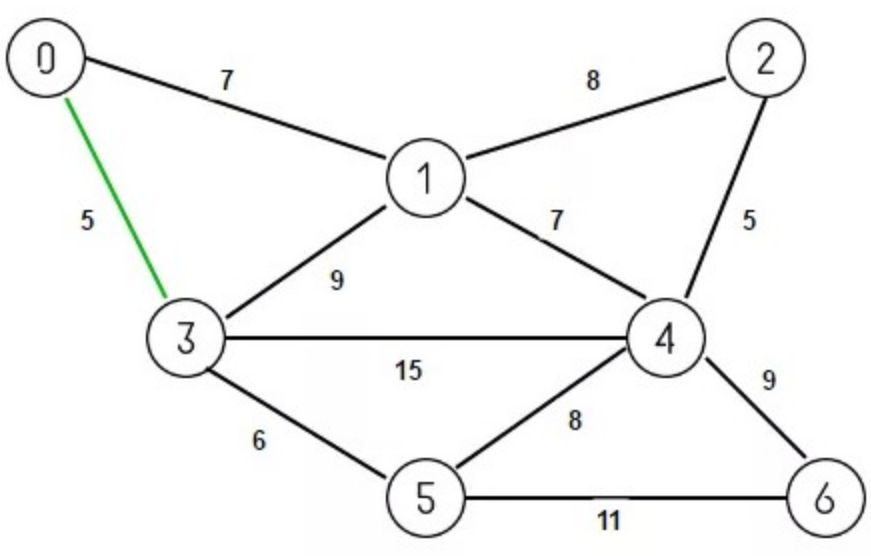
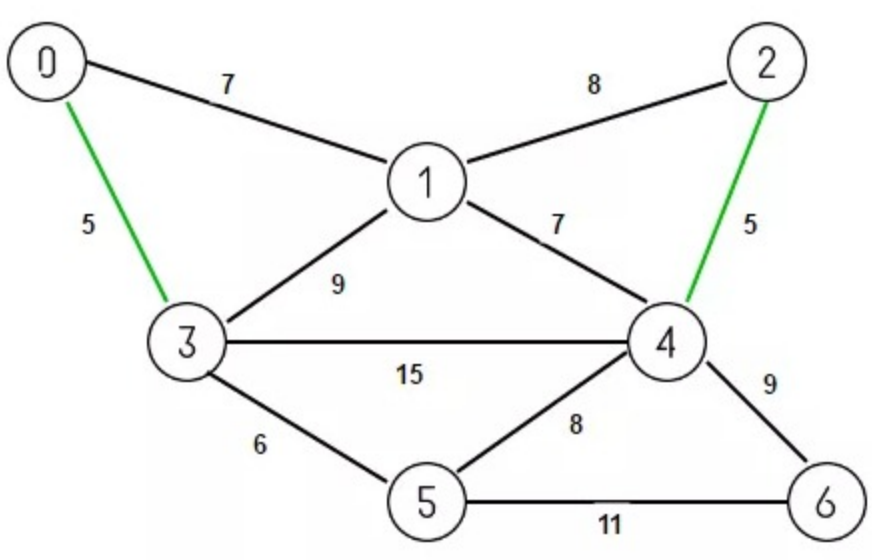
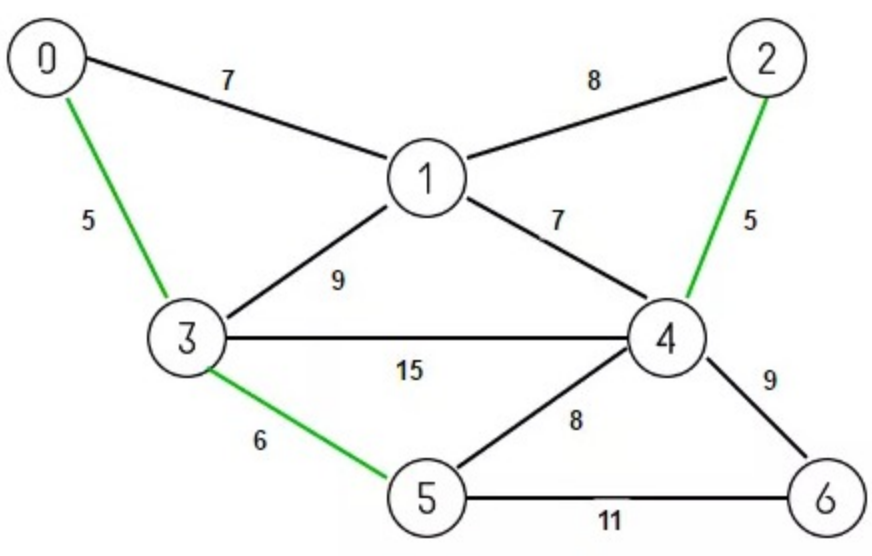
Let's illustrate this by taking example of above graph. Initially our MST consists of only the vertices of given graph with no edges. We start by considering smallest weighted edge 0-3 having weight 5. As no cycle is formed, we include it in our MST.



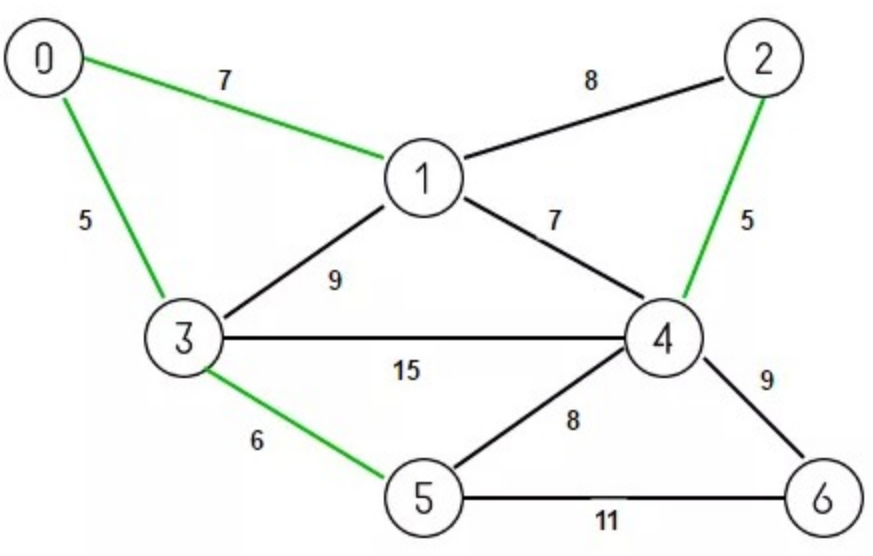
We next consider smallest weighted edge 2-4 also having weight 5. As no cycle is formed, we include it in our MST.



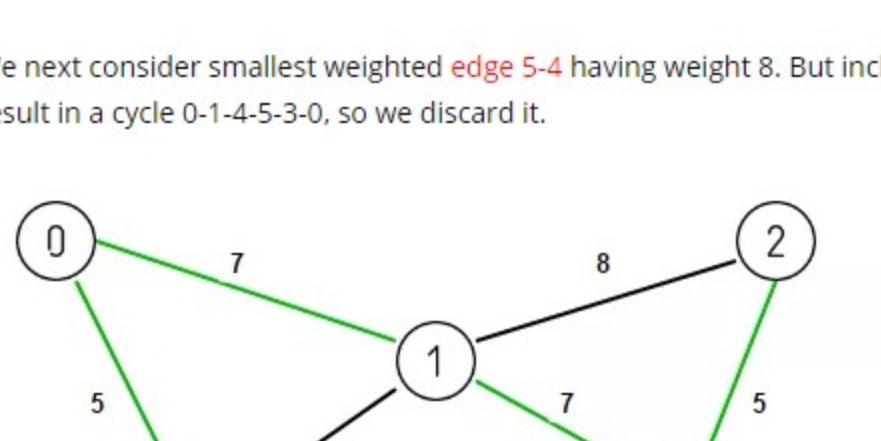
We next consider smallest weighted edge 3-5 having weight 6. As no cycle is formed, we include it in our MST.



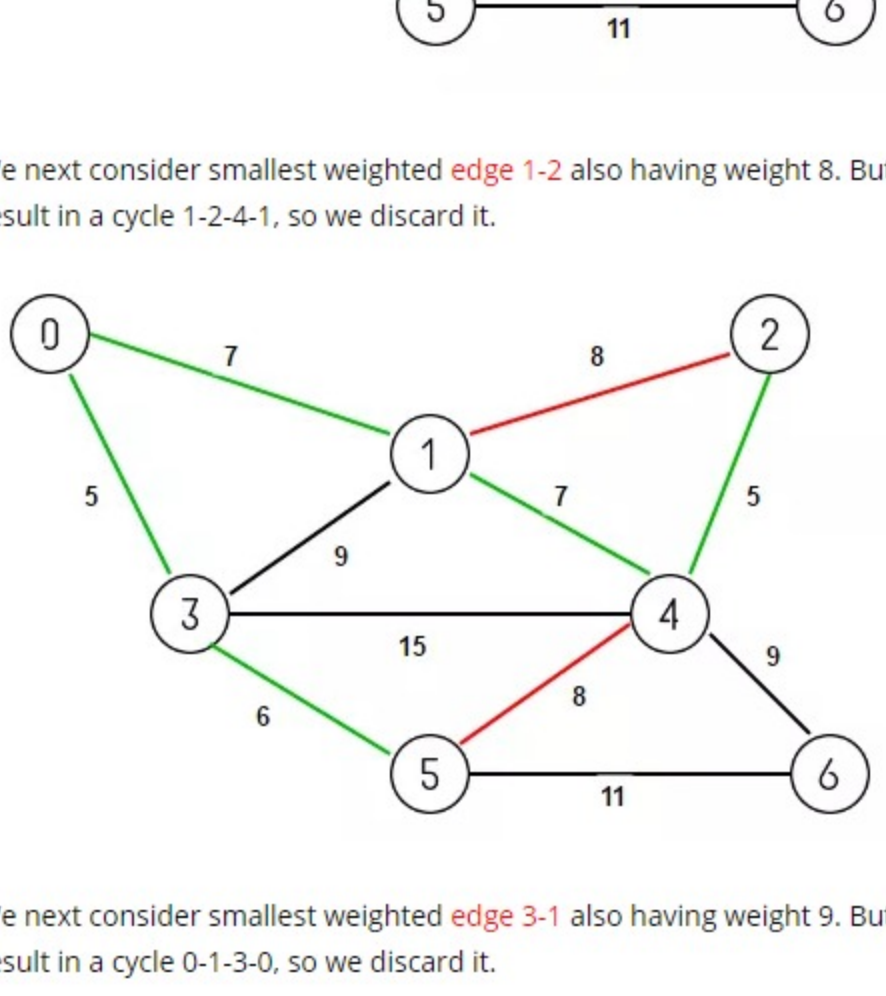
We next consider smallest weighted edge 1-4 also having weight 7. As no cycle is formed, we include it in our MST.



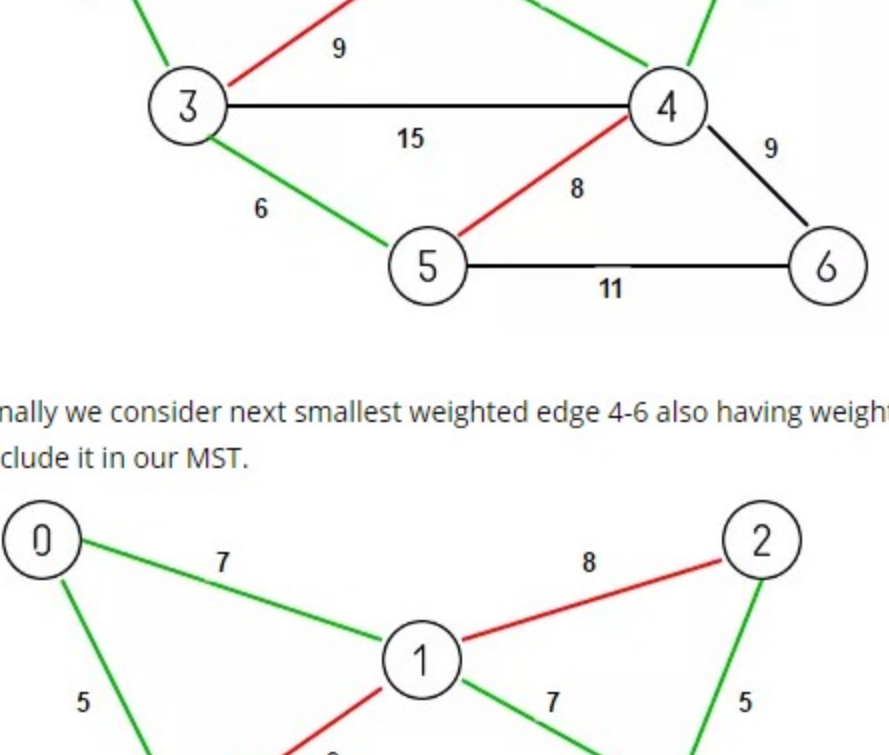
We next consider smallest weighted edge 5-4 having weight 8. But including this edge in MST will result in a cycle 0-1-4-5-3-0, so we discard it.



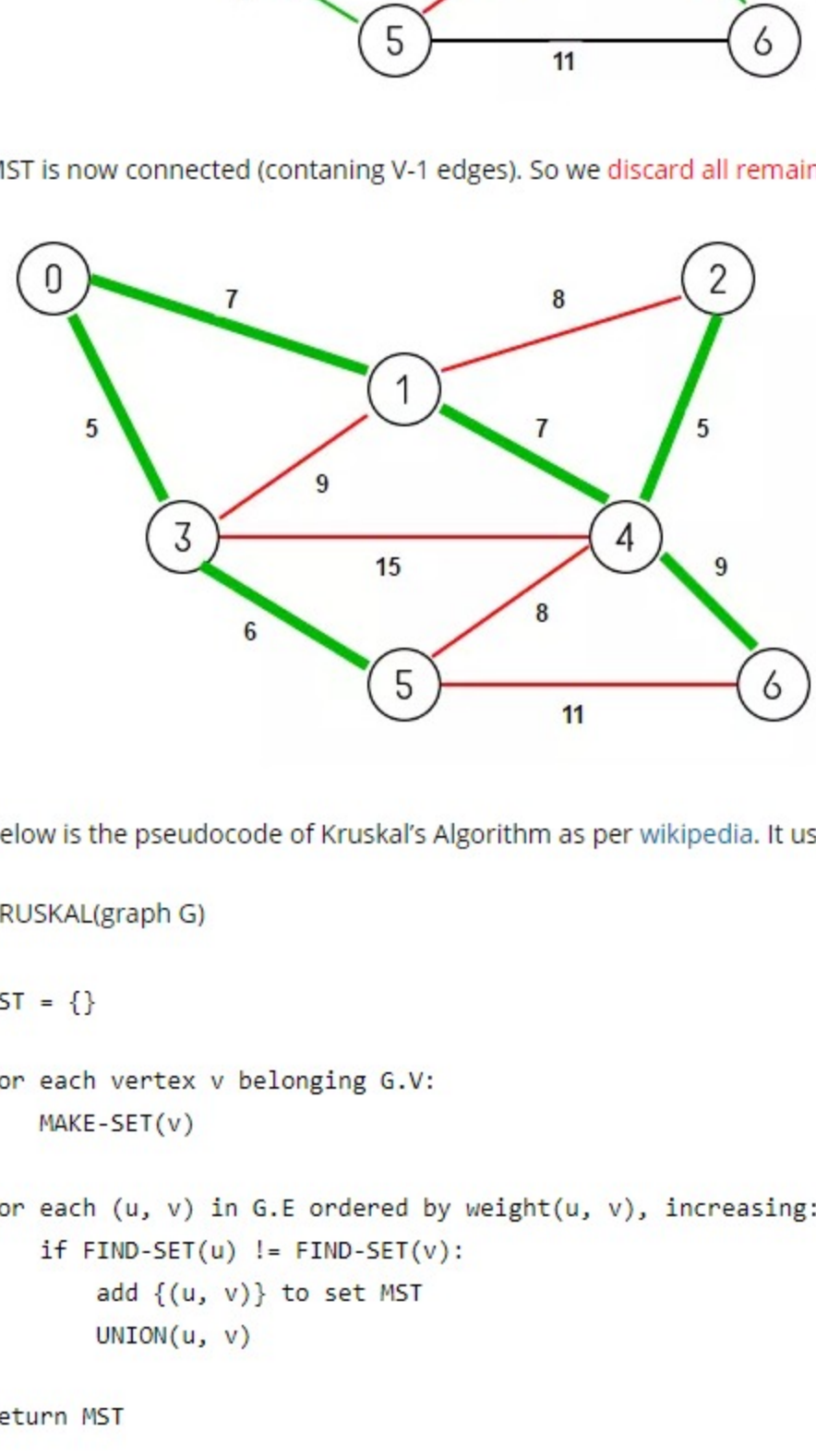
We next consider smallest weighted edge 1-2 also having weight 8. But including this edge in MST will result in a cycle 1-2-4-1, so we discard it.



We next consider smallest weighted edge 3-1 also having weight 9. But including this edge in MST will result in a cycle 0-1-3-0, so we discard it.



Finally we consider next smallest weighted edge 4-6 also having weight 9. As no cycle is formed, we include it in our MST.



MST is now connected (contaning V-1 edges). So we discard all remaining edges.

Below is the pseudocode of Kruskal's Algorithm as per wikipedia. It uses disjoint-set data structure.

KRUSKAL(graph G)

```
MST = {}

for each vertex v belonging G.V:
    MAKE-SET(v)

for each (u, v) in G.E ordered by weight(u, v), increasing:
    if FIND-SET(u) != FIND-SET(v):
        add {(u, v)} to set MST
        UNION(u, v)

return MST
```