COMP20230: Data Structures & Algorithms **2018-19 Semester2**

Lecture 18: April 10

Graph (3)

Lecturer: Dr. Andrew Hines *Group 8: Menghao Su and Renjie Fu*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

18.1 Outline

The lecture is mainly about MST which is short for the minimum spanning tree and two methods about it called Kruskal's algorithm and Prim's algorithm respectively. The minimum spanning tree is widely used in telecommunications and network, Tour operations, network of roads in cities, Nanoscale DNA assembly and so on. This lecture shows what a minimum spanning tree looks like(example in figure 1) and how to be implemented by Kruskal's algorithm and Prim's algorithm.

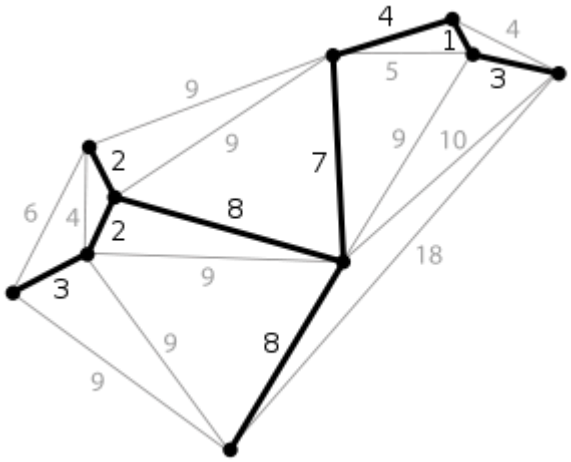


Figure 1

18.2 Definition of spanning tree

Before trying to know what a minimum spanning tree is, it is necessary to make sure what a spanning tree is and how to represent it(shown in Figure 2). A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. So, a spanning tree does not have cycles and it cannot be disconnected. Through this, we can find that for each undirected and connected Graph G, there will be at least one spanning tree, while a disconnected graph does not have even one since not all its vertices can be spanned.

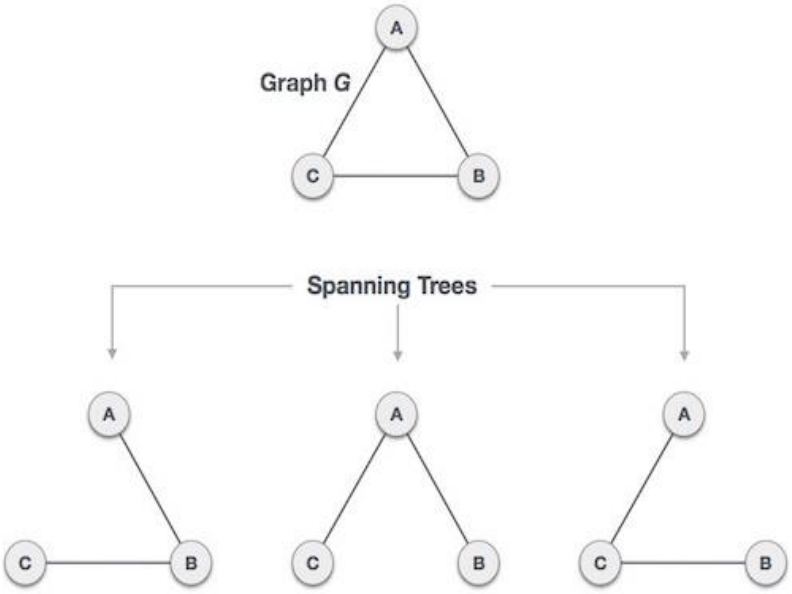


Figure 2

18.3 Minimum Spanning Tree

A minimum spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

Hence, we can summarize the key characters of MST here:

- It must be a connected edge-weighted undirected graph.
- All the vertices should be connected.
- There is no cycle in the graph.
- The total sum of edge weight(cost) of the path should be minimum.

18.3.1 Prim's Algorithm

Prim's Algorithm is one of the main algorithms to get the minimum spanning tree which is extremely to Dijkstra's algorithm(Dijkstra's algorithm is a shortest path algorithm to traverse weighted edges) and belong to Greedy Algorithm.

18.3.1.1 Procedure of Prim's Algorithm

As the most obvious character of Prim's Algorithm, it starts with a random node. The graphic explanation of Prim's Algorithm is shown below in Figure 3

1. One random vertex will be got in a graph and set as a tree root node.
2. Find the minimum-weighted edge among all edges which are connected to neighbours but not the edges that have already been in a tree. Then this edge we found should be added to the tree.
3. Go along the edge to the next vertex and set it as visited.
4. In the end, repeat the procedure from step 2 until we get the spanning tree, and that spanning tree will satisfy as a minimum spanning tree(n-1 edges have been added in the tree).

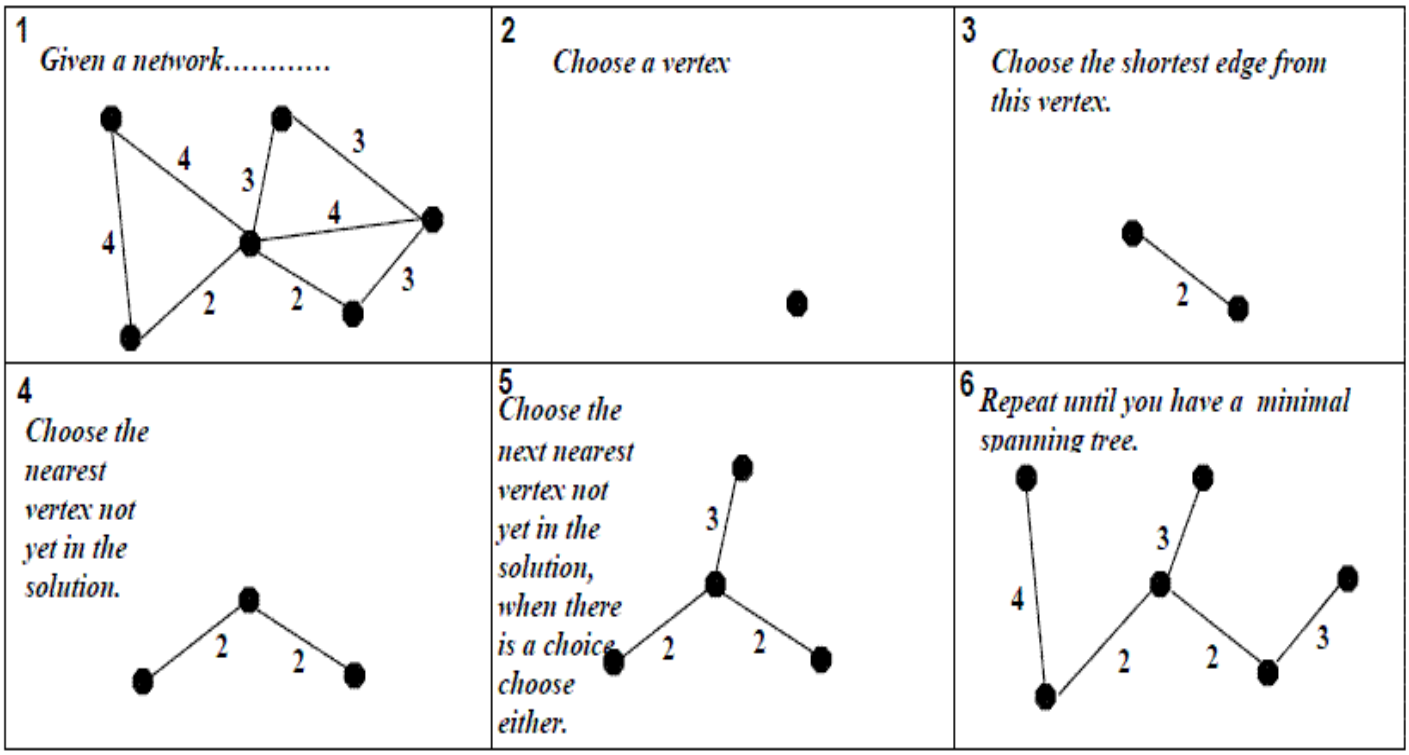


Figure 3

18.3.1.2 Prim's Algorithm Pseudocode

```
function Prim:
Input:  a connected undirected weighted graph G
Output:  T a minimum spanning tree based on G
T ← tree with all nodes from G but no edge
visited ← {random node}
while visited does not contain all nodes from G do
    minimum ← random edge (m,n) with m in visited and n not in visited
                                                    and no cycle
    for all node n not in visited reachable from a node m in visited do
        if weight of edge (m, n) < weight of minimum and no cycle
            minimum ← edge (m, n)
        endif
    endfor
    add n in minimum to visited
    add edge minimum to T
endwhile
return T
```

18.3.1.3 Complexity of Prim's Algorithm

Basically, when V mean vertex and E is edge, the complexity of Prim's algorithm is $O(|V|^2)$ but it can be improved to $O(|E|+|V|\log|V|)$ or $O(|E|\log|V|)$ after the optimization.

18.3.2 Kruskal's Algorithm

Another appropriate algorithm can be used to get a minimum spanning tree is called Kruskal's Algorithm. Similarly, it is also based on Greedy Algorithm. The strategy of Kruskal's Algorithm is not the same as that of Prim's Algorithm.

18.3.2.1 Procedure of Kruskal's Algorithm

Differently, Kruskal's Algorithm starts from the edge with the least weight as its most momentous character. The graphic explanation of Kruskal's Algorithm is shown below in Figure 4

1. Create a set containing all the edges in the graph.
2. Select the edge with the minimum weight in the set.
3. Add the edge into the span if the edge connects two different trees except the edge causes a cycling(connects back to the its own tree).
4. After that, repeat the same manipulation from step 2 and go over and over until we get the minimum spanning tree.

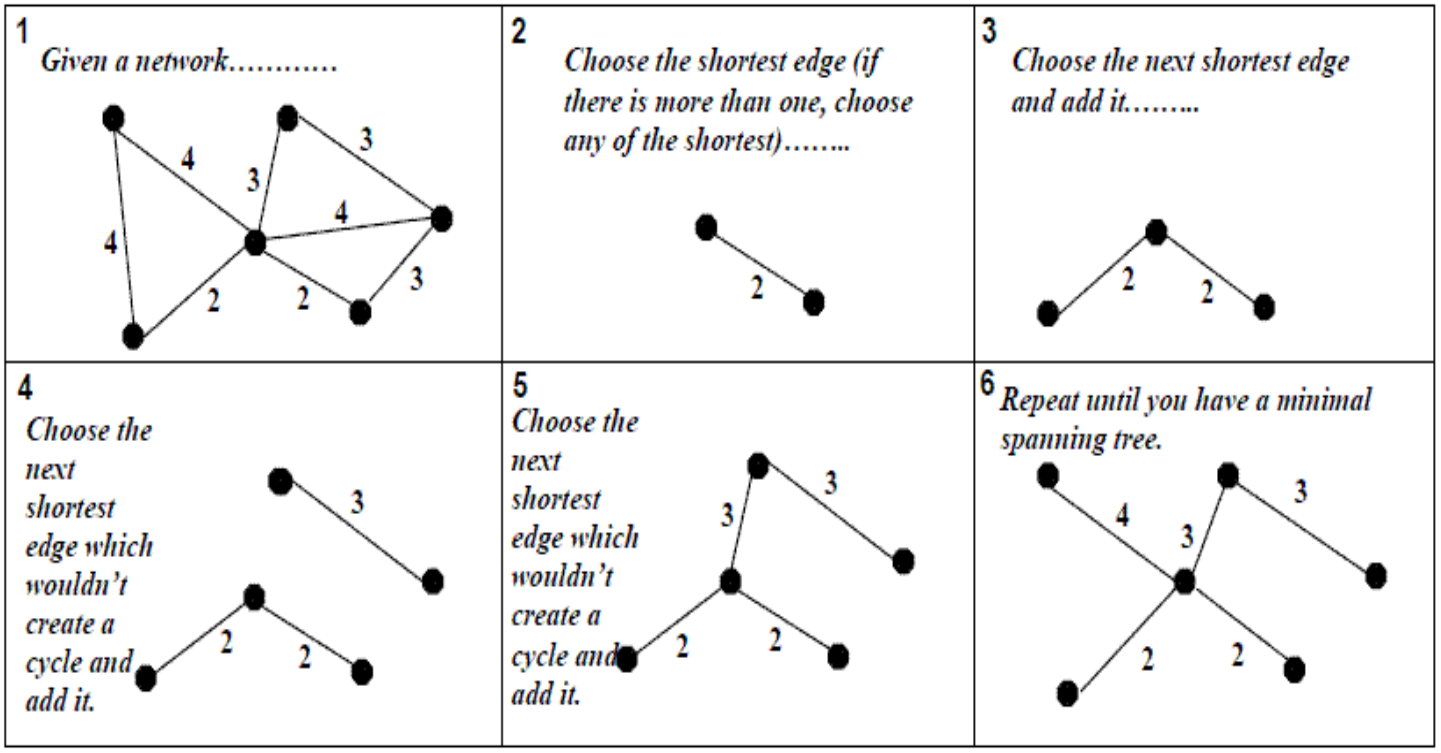


Figure 4

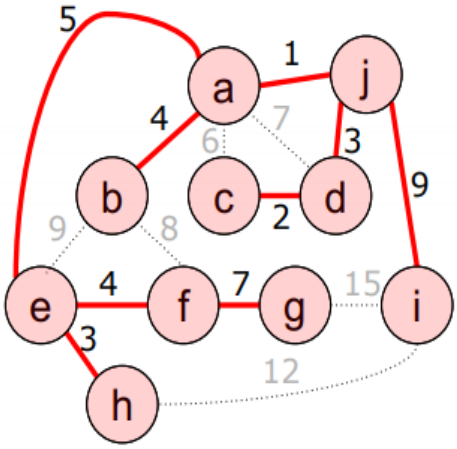
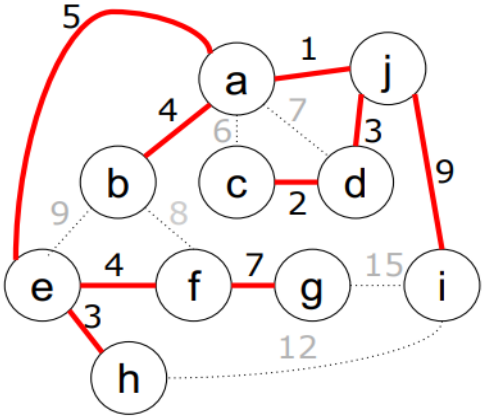
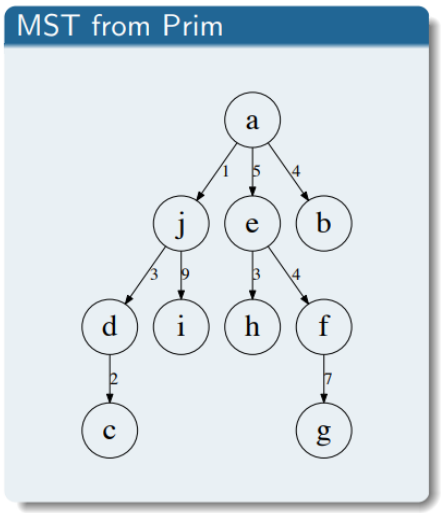
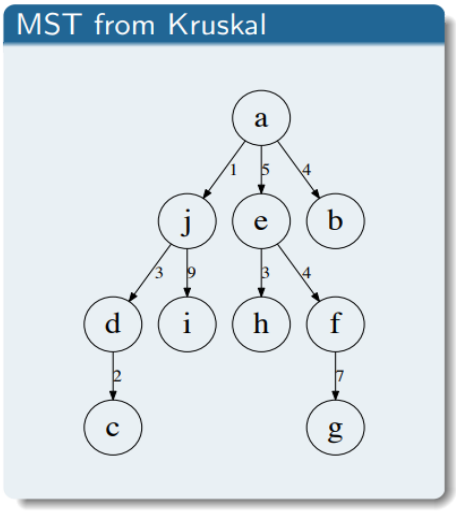
18.3.2.2 Kruskal's Algorithm Pseudocode

```
function Kruskal:
Input:  a connected undirected weighted graph G
Output:  T a minimum spanning tree based on G
T ← tree with all nodes from G but no edge
while T is not connected do
    chosen ← random edge from G not in T and not creating cycle
    for each edge e in G do
        if e is not in T and does not create a cycle in T
                                and weight e < weight chosen then
            chosen ← e
        endif
endfor
    add chosen to T
endwhile
return T
```

18.3.2.3 Complexity of Kruskal's Algorithm

Differently, when V mean vertex and E is edge, the complexity of Kruskal's algorithm is O(|E|log|E|), while the optimized complexity is O(|E|a(|V|))*.

18.4 Comparison

	Prim's Algorithm	Kruskal's Algorithm
Complexity	O(|V| ²)	O(|E|+|V|log|V|) or O(|E|log|V|)
Weighted, Undirected Graph		
MST		

18.5 Summary

Through this lecture, we learnt about the minimum spanning tree and two methods to get a minimum spanning tree. Prim's Algorithm and Kruskal's Algorithm were introduced in lecture, Prim's Algorithm focuses on starting with a random vertex, while Kruskal's Algorithm focuses on creating a new set with all edges inside and selecting the edge with minimum weight inside.

Besides, Dijkstra's Algorithm was also mentioned, which was really similar to Prim's Algorithm but is used to get a shortest path at a given vertex, while Prim's Algorithm and Kruskal's Algorithm are used to get the minimum spanning tree when comparing the application.

Reference

[1] https://www.tutorialspoint.com/data_structures_algorithms/spanning_tree.htm.

[2] https://www.quora.com/What-is-the-best-explanation-for-Prims-and-Kruskals-algorithms

[3] UCD Computer Science, Comp20230 Lecture Note, Dr Andrew Hines