Week 6

Matching*

6.1 Last Week Recap

Theorem: If G is a simple graph then diameter of (G) $\geq$ 3 therefore diam (G^c) $\leq$ 3
PROOF:
diam(G) $\geq$ 3

1. $\exists u, v, \ni v(G)$ subject to u and v are not adjacent.

2. u and v have no common neighbour.

u v does not exist in G $\forall x \in V(G) = \{u, v\}$
u v exist in G^c also ux or vx will also exist. All x nodes will be connected to either u or v

$$\forall x \in V(G) = \{u, v\}$$

therefore,
diam (G^c) $\leq$ 3

6.2 Theorem: Let G be a simple graph with diameter (G) at least 4 i.e. diam(G) $\geq$ 4 ; then Prove that diam (G^{c}) $\leq$ 2

we have to proove $p \Rightarrow q$
where p = diam(G) $\geq$ 4
q = diam (G^c) $\leq$ 2
Contrapositive Technique
$\sim q \Rightarrow \sim p$

*Lecturer: ANAND MISHRA. Scribe: MT19AIE217 - ARJIT SONI.

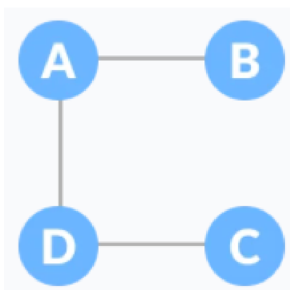
$\sim q = diam(G) \geq 3$
$\sim p = diam(G) \leq 3$
Since we knew that
$diam(G) \geq 3 \Rightarrow diam(G^c) \leq 3$

therefore,
$diam(G^c) \geq 3 \Rightarrow diam(G) \leq 3$
diam(G) ≤ 3

6.3 Spanning Tree

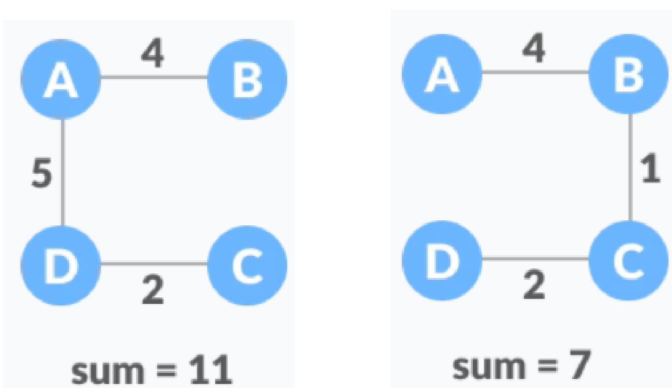
Definition: Spanning tree is without making a cycle which includes all the vertices of the graph with a minimum possible number of edges. If a vertex is missed, then it is not a spanning tree.

Figure 6.1: Example of a spanning tree



Minimum Spanning Tree: A minimum spanning tree is a spanning tree in which the sum of the weight of the edges is as minimum as possible.

Figure 6.2: Example of Minimum spanning tree



The minimum spanning tree from the above spanning trees is with weight 7.

Prim's Algorithm: This algorithm was suggested to find minimum spanning tree. It belongs to a category of algorithms known as greedy algorithms, which seek out the local optimum in the hopes of discovering a global optimum.
We begin with one vertex and gradually increase the weight of the edges until we attain our goal.
The following are the stages for implementing Prim's algorithm:

1. Create a random vertex to start the minimum spanning tree.
2. Find the minimum of all the edges that connect the tree to new vertices, then add it to the tree.
3. Step 2 should be repeated until we get a minimum spanning tree.

Figure 6.3: Started with Weighted Graph

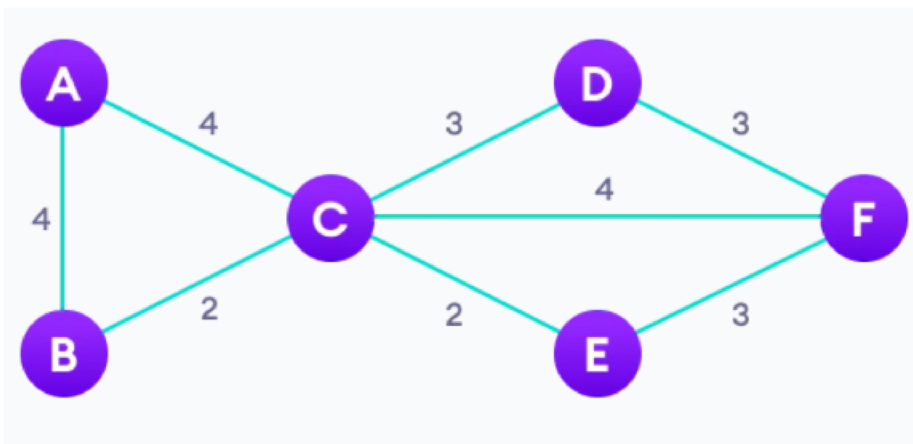
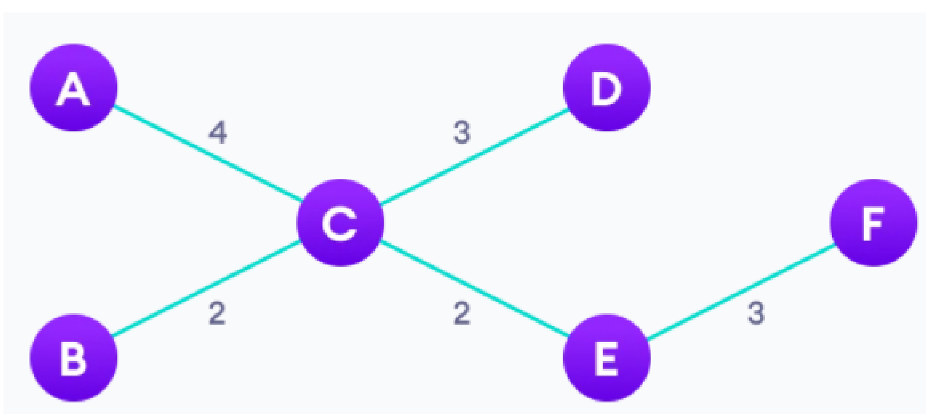


Figure 6.4: Repeat until you have a spanning tree



Kruskal's Algorithm: Kruskal's algorithm is a minimum spanning tree algorithm that takes a graph as input and discovers a subset of the network's edges that are not connected.

- It forms a tree that includes every vertex
- It has the minimum sum of weights among all the trees that can be formed from the graph

This method begins with the edges with the lowest weight and continues to add edges until the goal is reached.
The following are the steps for implementing Kruskal's algorithm:

1. Sort all of the edges from low to high.
2. Add the spanning tree to the edge with the heaviest weight. If adding the edge resulted in a cycle, discard it.
3. Continue to add edges until we've reached all of the vertices.

Figure 6.5: Started with Weighted Graph

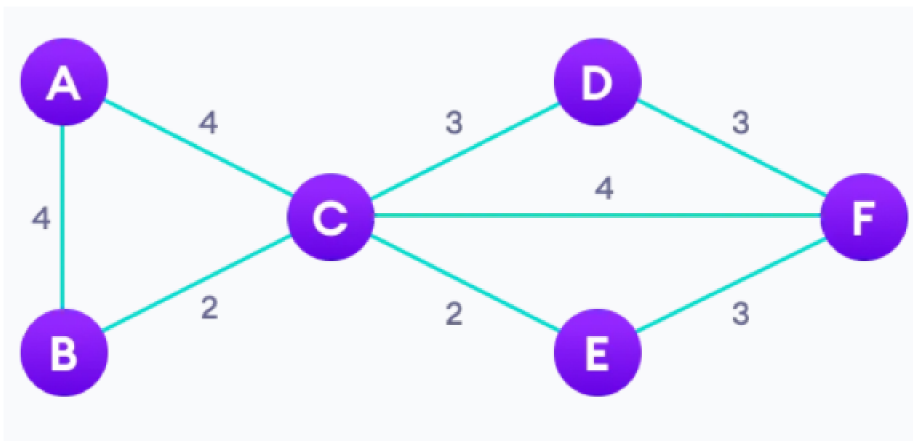
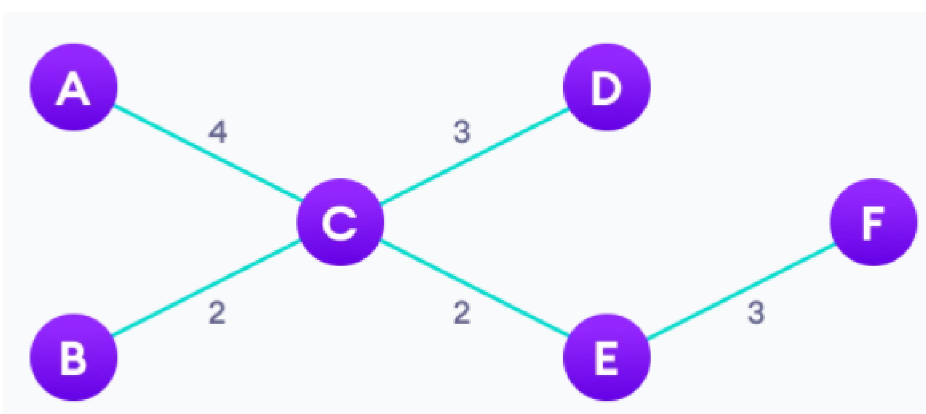


Figure 6.6: Repeat until you have a spanning tree



6.4 Matching and Covers

Matching is a set of non loop edges with no sharing end points.
Eg.

$\{AB, CD\} = M1$
$\{CD\} = M2$

Perfect Matching: A matching (M) of graph (G) is said to be a perfect match, if every vertex of graph g (G) is incident to exactly one edge of the matching (M). Every perfect matching of graph is also a maximum matching of graph, because there is no chance of adding one more edge in a perfect matching graph.

Maximum and Maximal Matching: A matching M of graph 'G' is said to maximal if no other edges of 'G' can be added to M.
It is also known as largest maximal matching. Maximum matching is defined as the maximal matching with maximum number of edges.
The number of edges in the maximum matching of 'G' is called its **matching number**.

If there are K components in a graph No of Perfect matching of a graph = No of Perfect matching C1 * no of perfect matching C2 * ....* No of perfect matching in Ck

6.5 Theorem: Every Tree has at most one perfect matching.

Take any vertex. If it's matches to the same vertex in both perfect matching, then it had degree zero on the symmetric difference. Otherwise it has degree two.
Second, after removing all isolated vertices in the symmetric difference, all vertices have degree two. Take any such vertex and follow its two edges. What you get is a growing path that eventually closed to a cycle since the graph is finite.

Since trees have no cycles, this implies that any two perfect matching are equal, by consisting their symmetric difference.

A different proof is by induction. The idea is that every leaf must be matched to its unique neighbor.
Proof by Induction: A tree having n=1 node has 0 Perfect matching for n=3, perfect matching = 0
Induction Hypothesis: Suppose the theorem is true for $n \leq K$ nodes.