Week 6

Diameter, Spanning Tree and Matching*

6.1 Diameter

6.1.1 If G is a simple graph then $\text{diam}(G) \geqslant 3 \rightarrow diam(Gc) \leq 3$.

Theorem6.1. If G is a simple graph then diam(G) $\geq 3 \rightarrow diam(G) \leq 3$.

Proof: $\text{diam}(G) \geqslant 3$
$\Rightarrow \exists u, v \in v(G)$ such that.
(i) u and v are not adjacent.
(ii) u and v have no common neighbours.
i.e uv does not exist in G and $Vx \in V(G) - \{u, v\}$, either ux or vx does not exist in G.
Now, for G^c, $\Rightarrow uv\cdot$ exist in G^c and ux or vx will also exist in G^c.

In Figure 6.1.

$\forall x' \in V(G) - \{u, v\}$
from any node we start, we can reach from ore node to another node in ≤ 3 steps.
Therefore diam(G^c) ≤ 3

6.1.2 If G is a simple graph then $\text{diam}(G) \geqslant 4 \rightarrow \text{diam}(G) \leqslant 2$

Proof:
Statement $P : \text{diam}(G) \geqslant 4$
Statement $Q : \text{diam}\ (G^c) \leqslant 2$
We have to prove that $P \Rightarrow Q$
We will prove equivalent to above

*Lecturer: Dr. Anand Mishra. Scribe: Sarthak Kumar Singh.

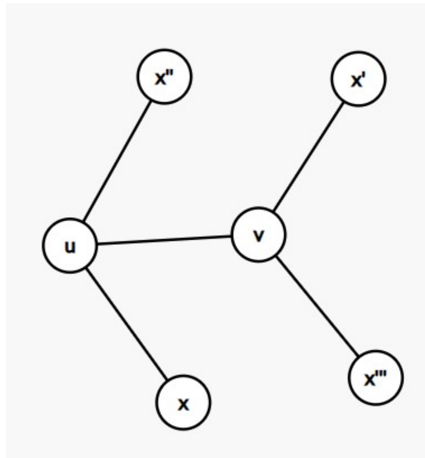

Figure 6.1: Graph of G^c

i.e. $\sim Q \Rightarrow \sim P$
$\sim Q \equiv \operatorname{diam}(G^c) \geqslant 3$
$\sim P \equiv \operatorname{diam}(G) \leqslant 3$
So, we have to prove here that $\operatorname{diam}(G) \geqslant 3 \Rightarrow \operatorname{diam}(G^c) \leqslant 3$
(replacing G with G^c)
$\operatorname{diam}(G^c) \geqslant 3 \Rightarrow$
$\operatorname{diam}((G^c)^c) \leqslant 3 \Rightarrow \operatorname{diam}(G) \leqslant 3$ (This we have proved in the above theorem.)

6.2 Spanning Tree

Definition: A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected.

6.2.1 Minimum Spanning Tree

Minimum spanning tree can be defined as the spanning tree in which the sum of the weights of the edge is minimum. The weight of the spanning tree is the sum of the weights given to the edges of the spanning tree.

6.2.2 Prim's Algorithm

It is a greedy algorithm that is used to find the minimum spanning tree from a graph. Prim's algorithm finds the subset of edges that includes every vertex of the graph such that the sum of the weights of the edges can be minimized.

Prim's algorithm starts with the single node and explores all the adjacent nodes with all the connecting edges at every step. The edges with the minimal weights causing no cycles in the graph got selected.

6.2.3 Krushkal's Algorithm

It is used to find the minimum spanning tree for a connected weighted graph.

How It works:
(i) First, sort all the edges from low weight to high.
(ii) Now, take the edge with the lowest weight and add it to the spanning tree. If the edge to be added creates a cycle, then reject the edge.
(iii) Continue to add the edges until we reach all vertices, and a minimum spanning tree is created.

6.3 Matching

Definition: Matching is a set of Non-loop edges with no shared end points.

In the diagram below:

$$\text{Matching } \Rightarrow \begin{matrix} M_1 = \{AB, CD\} \\ M_2 = \{BC\} \end{matrix}$$

6.3.1 Maximal and Maximum Matching

Maximum Matching: The maximum size matching or the Matching where you can not add one more matching.
Maximal Matching: It is matching that can not be enlarged by adding an edge.

for Example:
In the graph below,
Maximum matching $\Rightarrow \{AB, CD\}$ Maximal matching $\Rightarrow \{B, C\}$

6.3.2 Perfect Matching

Perfect Matching of a Graph having k connected components is the product of perfect Matching in all the connected components.

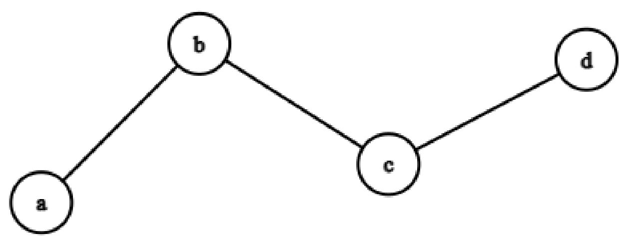

Figure 6.2: Example graph for Matching

6.3.3 Every Tree has atmost one Perfect Matching

Proof:- By induction
Base condition - A tree having $n = 1$ node has No. of PM=0, for $n = 2$ node, No. of PM=1
Induction hypothesis:-Suppose the theorem is True for $n \leqslant k$ nodes, Then we have to prove that it is also true for $n = k + 1$ nodes.
Suppose, we have added 1 more mode to the tree having k nodes. And then We break this extra node along with it's parent node as a Separate component, let's called this component as T_1 and the other component of the Tree T_2.
Now,
No. of nodes in $T_1 \leqslant k$ So, for T_1, No. of PM $\leqslant 1$
Also, No. of nodes in $T_2 \leqslant k$ So, for T_2, No. of PM $\leqslant 1$
So, for The whole Tree,
No. of PM = No. of PM(T_1)x$No. of PM(T_2)$
$No. of PM \leqslant 1$

6.3.4 Hall's Marriage theorem

An x-y Bigraph G has a matching that Saturates x iff $|N(S)| \geqslant |S| \forall S \in x$
Here $N(s) \subseteq Y$ is a set of neighbours of elements in S