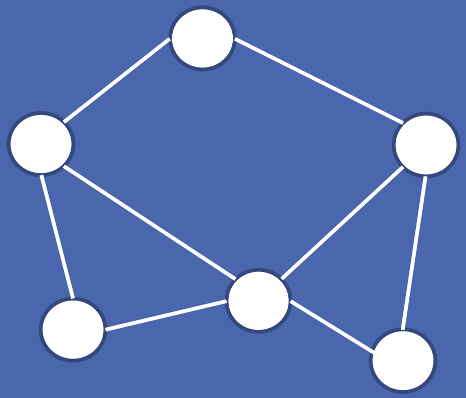
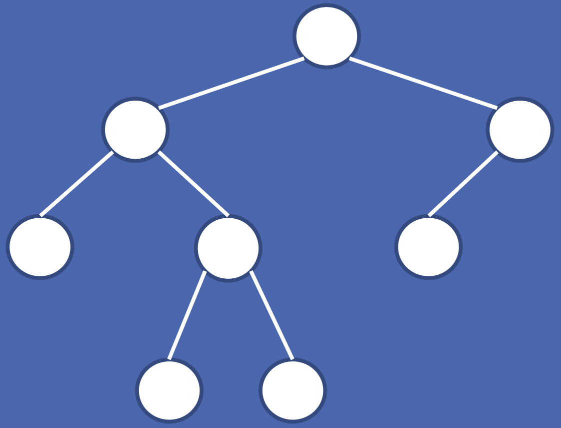
CS1231S TUTORIAL #11

Graphs and Trees





Learning objectives of this tutorial

Graphs and Trees

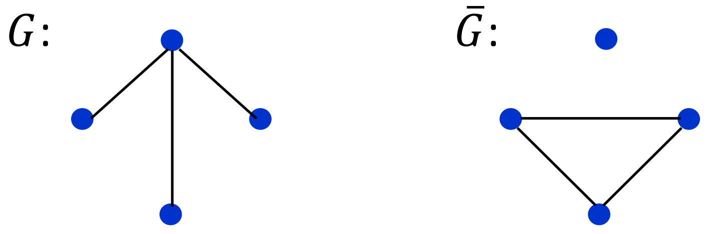
- Introducing complement graphs and self-complementary graphs.
- Calculating the number of walks of a certain length in a graph.
- Modeling a problem as a graph problem.
- Understanding pre-order, in-order and post-order traversals.
- Calculating the number of spanning trees in a graph.
- Applying Kruskal’s Algorithm and Prim’s Algorithm.

Definitions

If G is a simple graph, the complement of G, denoted $\bar{G}$, is obtained as follows: the vertex set of $\bar{G}$ is identical to the vertex set of G. However, two distinct vertices v and w of $\bar{G}$ are connected by an edge iff v and w are not connected by an edge in G.

A self-complementary graph is isomorphic with its complement.

A simple circuit (cycle) of length 3 is called a triangle.



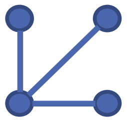
A graph G and its complement $\bar{G}$.

Q1. Draw all self-complementary graphs with

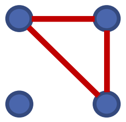
Complement graph

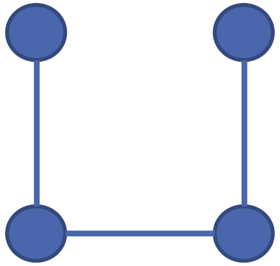
(a) 4 vertices

How about this?

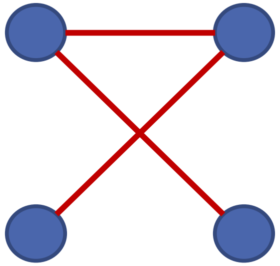


$\ncong$

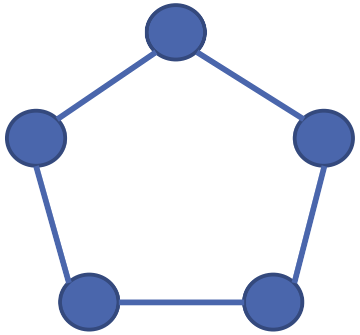




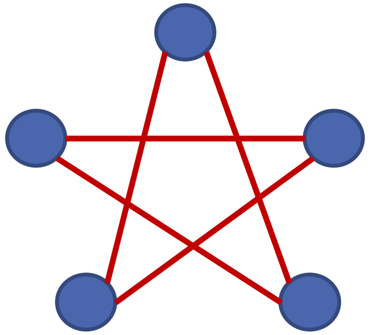
$\cong$

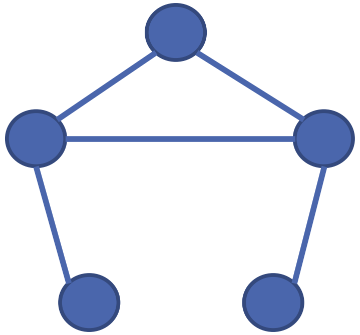


(b) 5 vertices

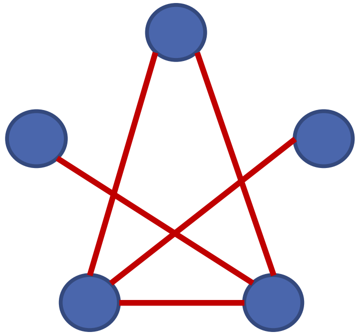


$\cong$





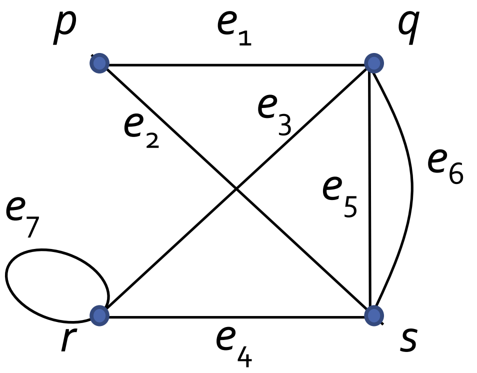
$\cong$



Q2. Show that every simple graph with at least 2 vertices has two vertices of the same degree.

1. Let G be a simple graph with $n \geq 2$ vertices.
2. Case 1: If G has no vertex with degree $n - 1$
   - 2.1 Then all vertices in G have degree lying in the range 0 to $n - 2$ inclusive.
3. Case 2: If G has a vertex v with degree $n - 1$
   - 3.1 Firstly, v does not have degree 0 (since $n - 1 > 0$).
   - 3.2 Also, as v has degree $n - 1$, it is connected to every other vertex, hence no other vertex can have degree 0 too.
   - 3.3 Hence, all vertices in G have degree lying in the range 1 to $n - 1$ inclusive.
4. In all cases, there are at most $n - 1$ possible vertex degrees among the n vertices.
5. Therefore, at least two vertices must have the same degree (by the Pigeonhole Principle).

Q3. Given this graph



(a) Write the adjacency matrix A for the graph. Let the rows and columns be p, q, r, s.

$$A = \begin{array}{c} \begin{array}{cccc} p & q & r & s \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ p \\ q \\ r \\ s \end{array}$$

(b) Find A^2 and A^3.

$$A^2 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 6 & 3 & 2 \\ 2 & 3 & 3 & 3 \\ 2 & 2 & 3 & 6 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 6 & 3 & 2 \\ 2 & 3 & 3 & 3 \\ 2 & 2 & 3 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 6 & 8 \\ 8 & 9 & 11 & 17 \\ 6 & 11 & 9 & 11 \\ 8 & 17 & 11 & 9 \end{bmatrix}$$

(c) How many walks of length 2 are there from p to q? From s to itself?

2 walks from p to q: $< pe_2se_5q >, < pe_2se_6q >$.

6 walks from s to s: $< se_2pe_2s >, < se_4re_4s >, < se_5qe_5s >,, < se_6qe_6s >, < se_5qe_6s >,, < se_6qe_5s >$.

(d) How many walks of length 3 are there from r to s? From s to p?

11 walks from r to s:

2 ways via q: $< re_3qe_1pe_2s >, < re_3qe_3re_4s >$.

3 ways via r: $< re_7re_7re_4s >, < re_7re_3qe_5s >, < re_7re_3qe_6s >$.

6 ways via s: $< re_4se_2pe_2s >, < re_4se_4re_4s >, < re_4se_5qe_5s >,$
$< re_4se_6qe_6s >, < re_4se_5qe_6s >, < re_4se_6qe_5s >$.

8 walks from s to p:

2 ways via p: $< se_2pe_2se_2p >, < se_2pe_1qe_1p >$.

4 ways via q: $< se_5qe_5se_2p >, < se_5qe_6se_2p >,$
$< se_6qe_5se_2p >, < se_6qe_6se_2p >$.

2 ways via r: $< se_4re_4se_2p >, < se_4re_3qe_1p >$.

Q4. Prove that for any simple graph G with 6 vertices, G or its complementary graph $\bar{G}$ contains a triangle.

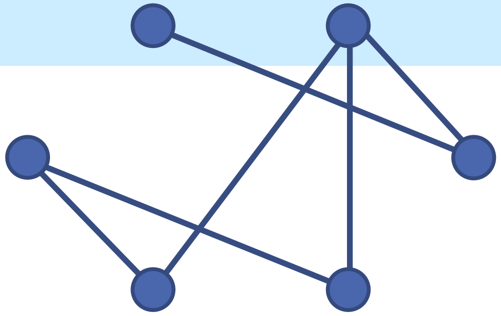


Figure 4a: Graph G

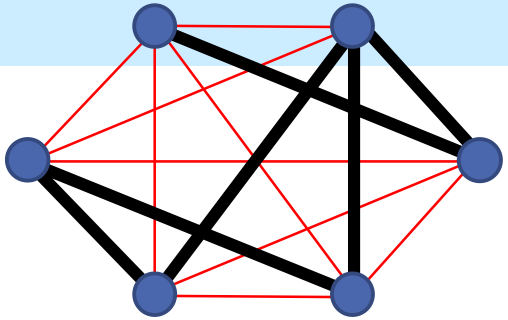


Figure 4b: Graph G'

1. Take any simple graph G with 6 vertices (figure 4a).
2. Draw a black edge between adjacent vertices in G and draw a red edge between non-adjacent vertices in G.
3. Call this graph G’ (figure 4b). Now G’ is a complete graph with every edge either black or red, and we want to prove that it has a black triangle or a red triangle.
4. Let v be an arbitrary vertex of G’.
   4.1 There are 5 edges incident to v, which are either black or red.
   4.2 Therefore, (at least) 3 of these 5 edges are of the same colour c (by Generalized PHP).
   4.3 For these 3 edges of colour c, name the vertices at the other end of these edges u_1, u_2, u_3 (figure 4c).
   4.4 Case 1: If there is an edge of colour c between any two of u_1, u_2, u_3, then that edge forms a triangle of colour c with the two edges coming from v. (In figure 4d, the 3 dashed lines are the edges between u_1, u_2, u_3. (In this example, two of them – $\{u_1, u_3\}$ and $\{u_2, u_3\}$ – are of colour c (red). Let’s pick $\{u_1, u_3\}$. We have a triangle of colour c with v, u_1, u_3.)
   4.5 Case 2: If there are no edges of colour c between any pair of u_1, u_2, u_3. Then the edges between these 3 vertices form a triangle of colour opposite to c.
5. In all cases, there is a triangle of the same colour.

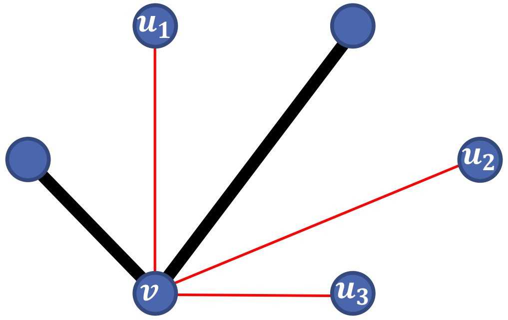


Figure 4c

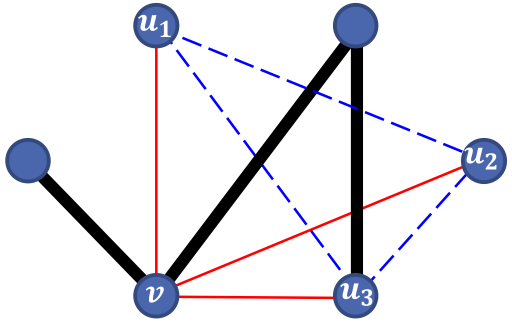
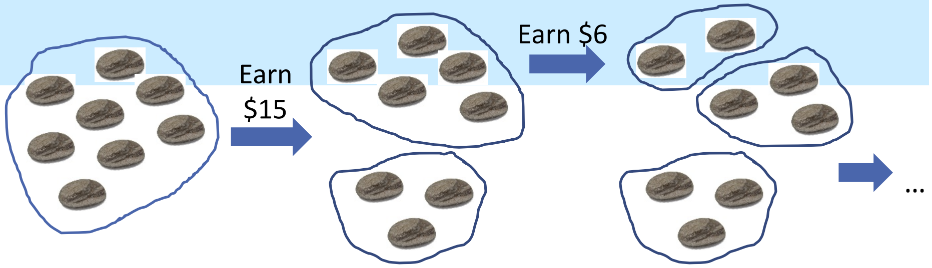


Figure 4d

Q5.

Given a pile of stones. At each step, you can separate a pile of k stones into two piles of k_1 and k_2 stones ($k_1 + k_2 = k$). On doing this, you earn \$($k_1 \times k_2$).

What is the maximum amount of money you can earn at the end if you start with a pile of n stone?

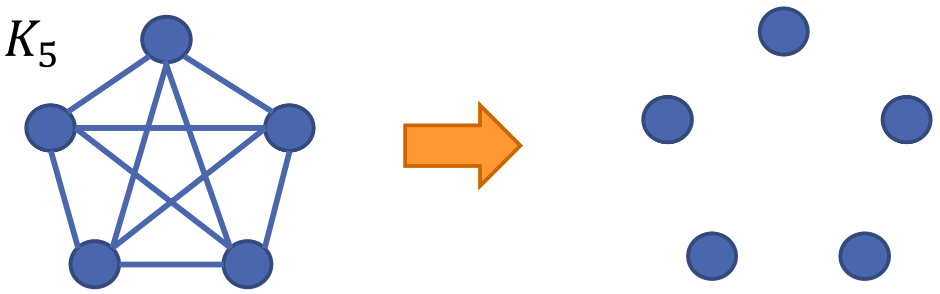


Model this as a graph problem. Let vertices represent the stones. Draw an edge between two vertices if the stones they represent are in the same pile.

In the beginning, we have a K_n graph.

In the end, we have a graph of n isolated vertices.

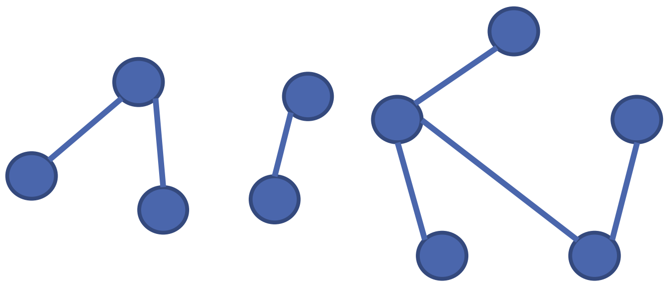
Eg: K_5



Separating a pile of k stones into two piles of k_1 and k_2 stones corresponds to removing $k_1 \times k_2$ edges from K_k.

A K_n graph has $\binom{n}{2} = \frac{n(n-1)}{2}$ edges. Therefore, the maximum amount one can earn is \$ $\frac{n(n-1)}{2}$.

Q6. How many edges are there in a forest with v vertices and k components?



1. Let the i^{th} component of the forest contains v_i vertices.

2. Each component is a tree, therefore the i^{th} component has $v_i - 1$ edges.
   (by Theorem 10.5.2)

3. Hence, the total number of edges in the forest is

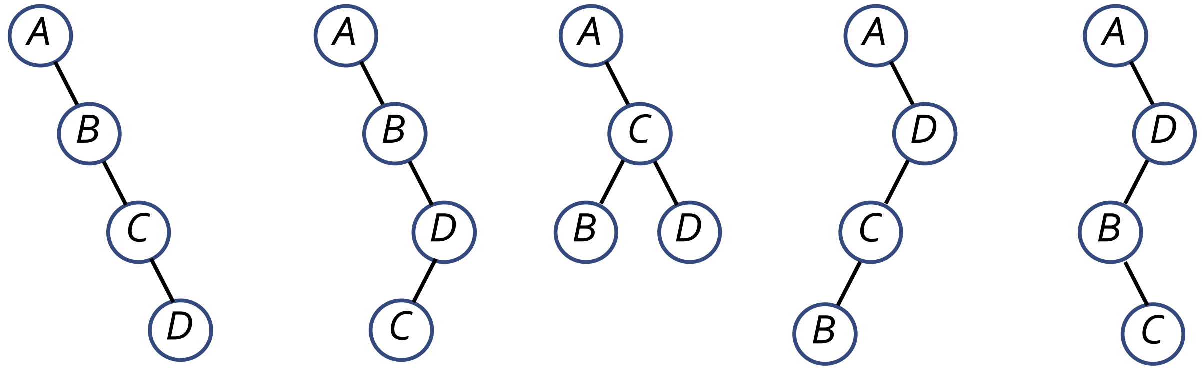
$$\sum_{i=1}^{k}(v_i - 1) = \sum_{i=1}^{k} v_i - \sum_{i=1}^{k} 1 = \boldsymbol{v} - k.$$

Theorem 10.5.2
A tree with n vertices ($n > 0$) has $n - 1$ edges.

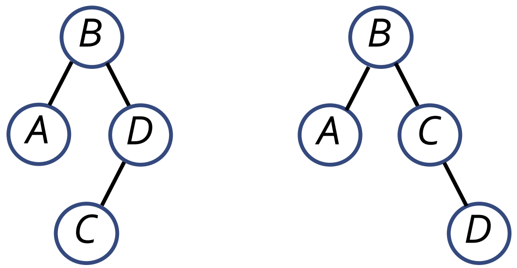
Q7. How many possible binary trees with 4 vertices *A*, *B*, *C* and *D* have this in-order traversal: *A B C D*? Draw them.

Total: 14

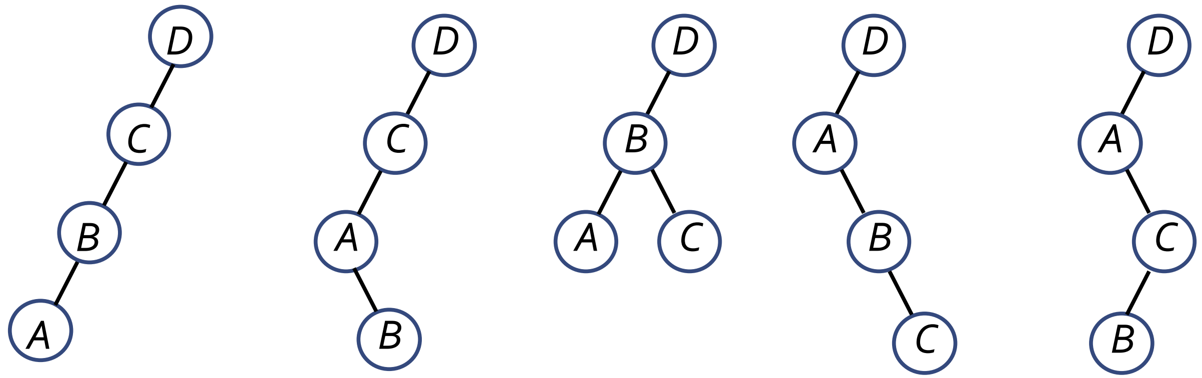
Rooted at *A*: 5



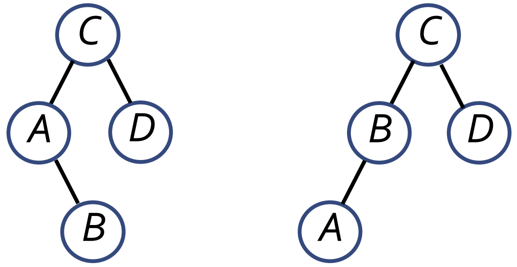
Rooted at *B*: 2



Rooted at *D*: 5

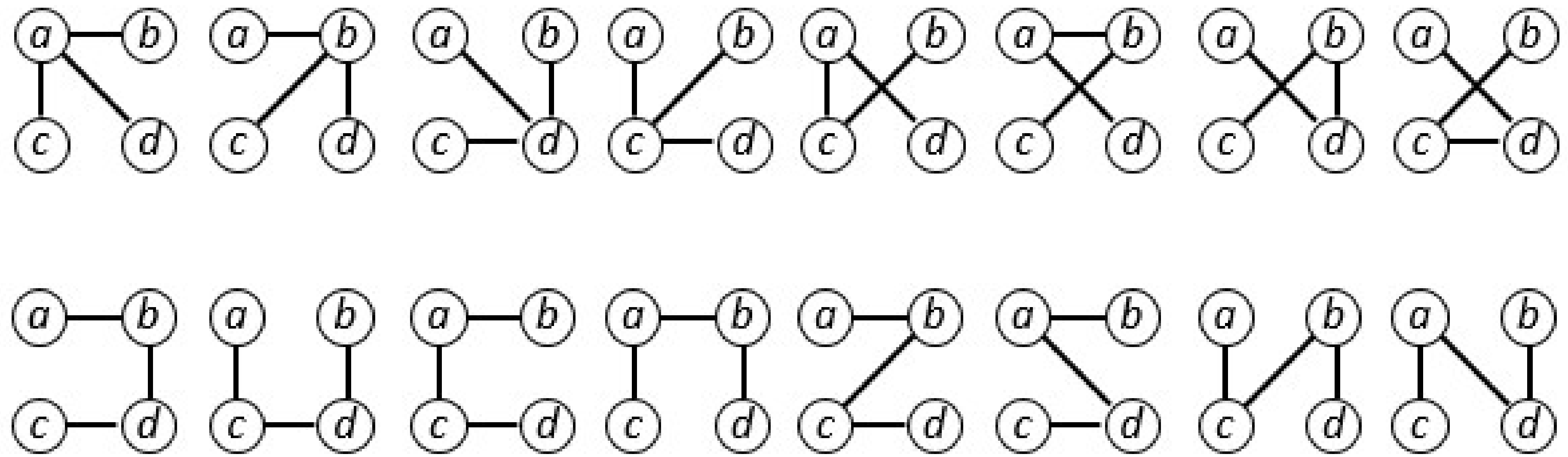


Rooted at *C*: 2

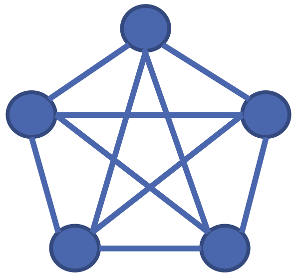


Q8. (a) The number of spanning trees in a complete graph K_n is n^{n-2}. Show all the spanning trees in K_4 below.

$4^{4-2} = 16$



Q8. (b) Following the given steps, verify that the number of spanning trees in K_5 is $5^{5-3} = 125$.



Adjacency matrix A for K_5:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Diagonal matrix D:

$$D = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}$$

$$M = D - A = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

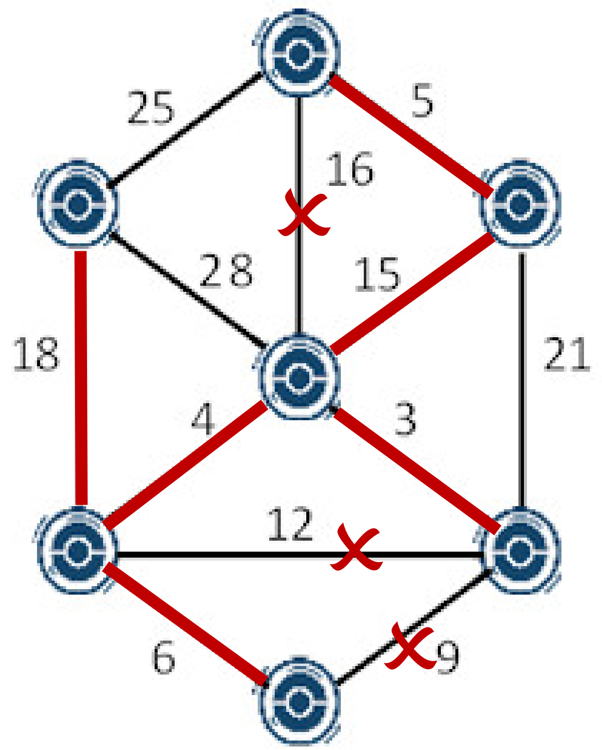
$$M' = \begin{pmatrix} 4 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & 4 \end{pmatrix}$$

$$\det(M') = 4\begin{vmatrix} 4 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 4 \end{vmatrix} + \begin{vmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & 4 \end{vmatrix} - \begin{vmatrix} -1 & 4 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & 4 \end{vmatrix} + \begin{vmatrix} -1 & 4 & -1 \\ -1 & -1 & 4 \\ -1 & -1 & -1 \end{vmatrix}$$

$$= 4(50) + (-25) - (25) + (-25) = \mathbf{125}.$$

Q9. Find the MST of the graph below.

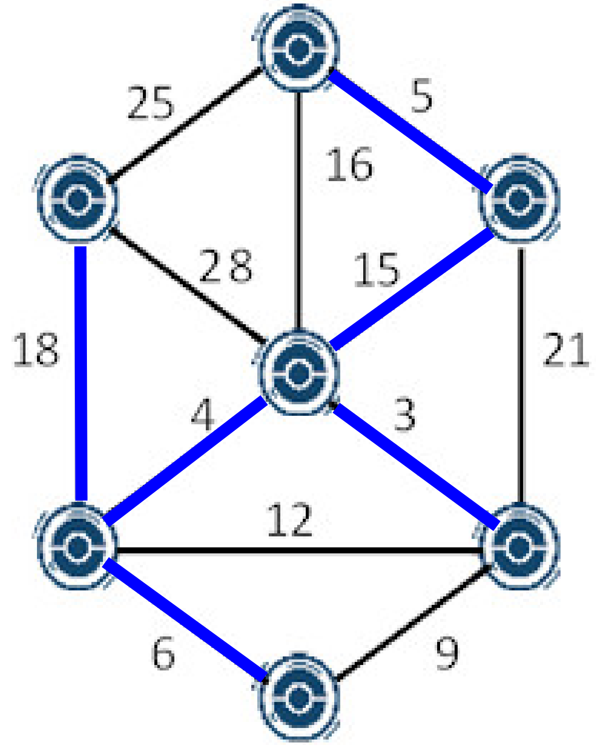
Kruskal's algorithm



Edges in non-decreasing order:
3
4
5
6
9
12
15
16
18
21
25
28

Weight of MST = $3 + 4 + 5 + 6 + 15 + 18 = 51.$

Q9. Find the MST of the graph below.



Prim's algorithm

Start from top vertex.

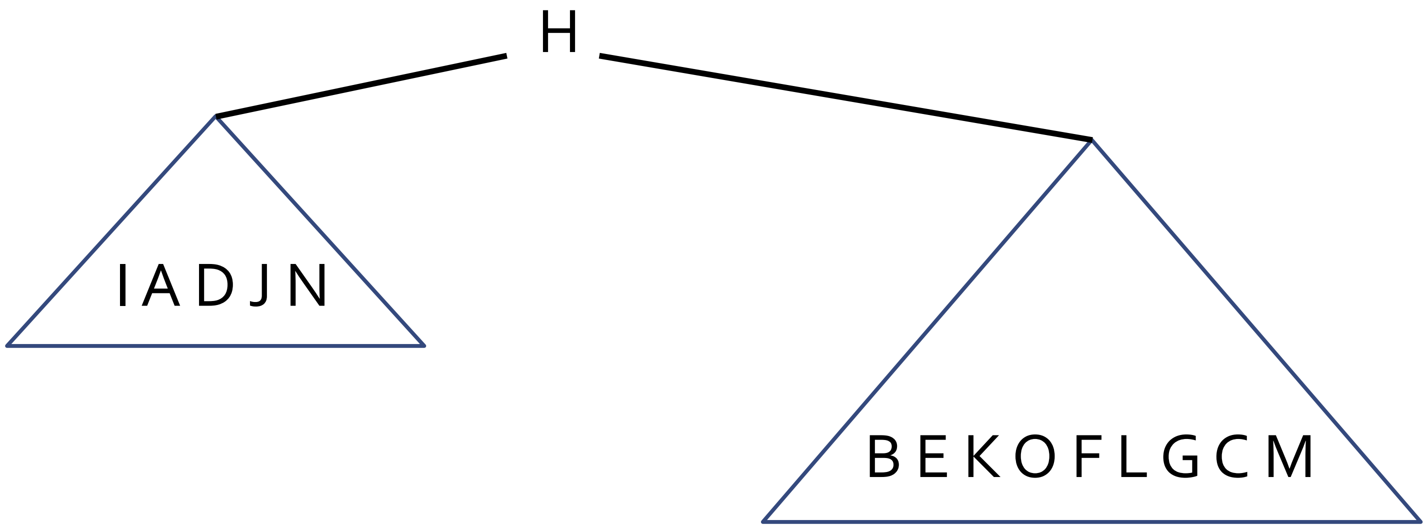
1. $\{5,16,25\} \rightarrow 5$
2. $\{25,16,15\} \rightarrow 15$
3. $\{25,28,4,3,21\} \rightarrow 3$
4. $\{25,28,4,12,9\} \rightarrow 4$
5. $\{25,28,18,6,9\} \rightarrow 6$
6. $\{25,28,18\} \rightarrow 18$

Weight of MST = $5 + 15 + 3 + 4 + 6 + 18 = 51$.

Q10. Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

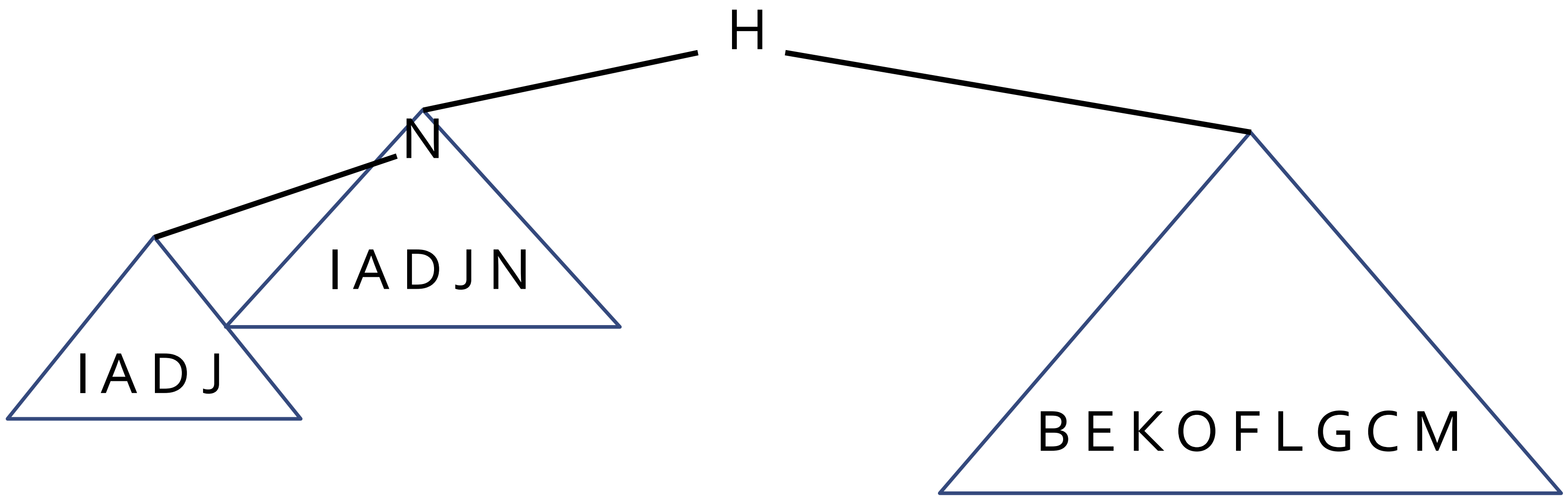
Pre-order: H N A I J D O B K E C L F G M



Q10. Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M

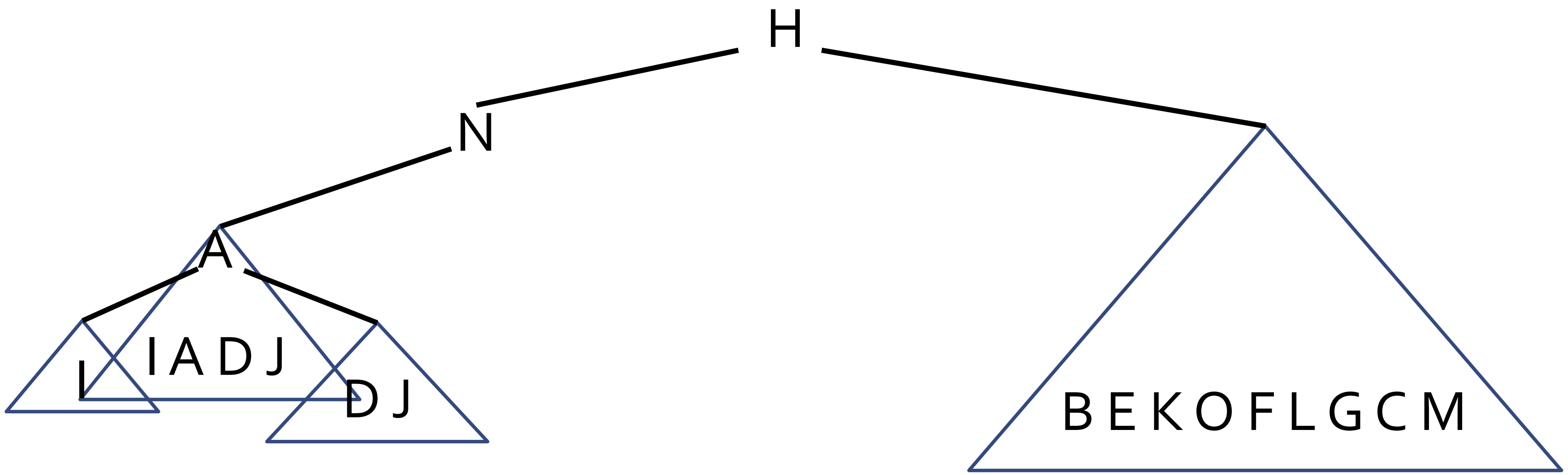


Q10.

Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M

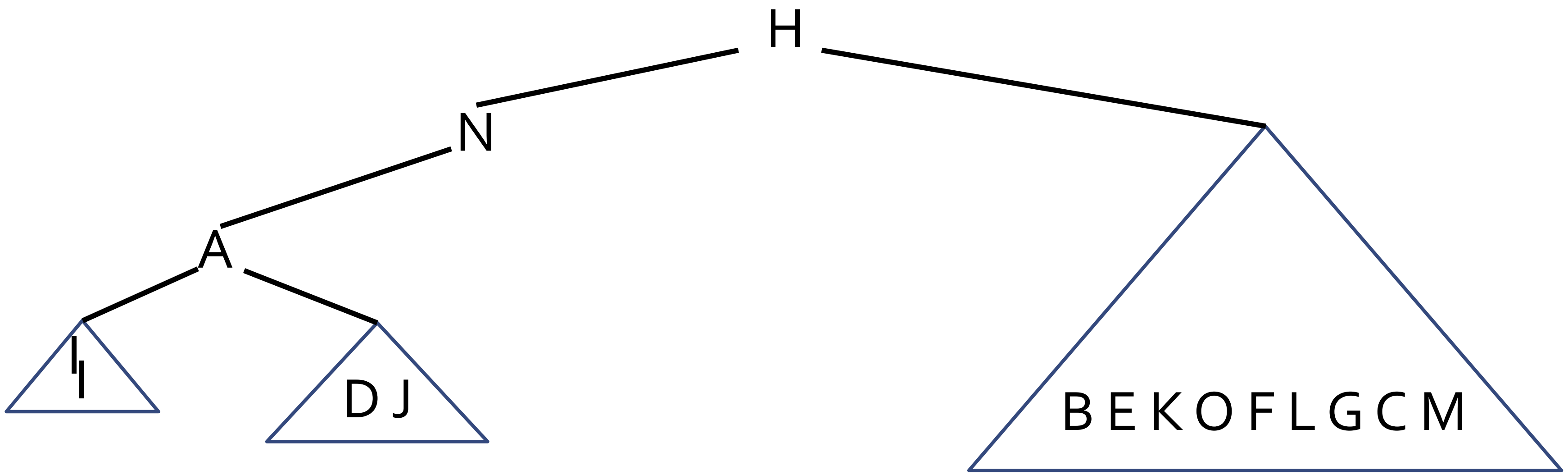


Q10.

Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

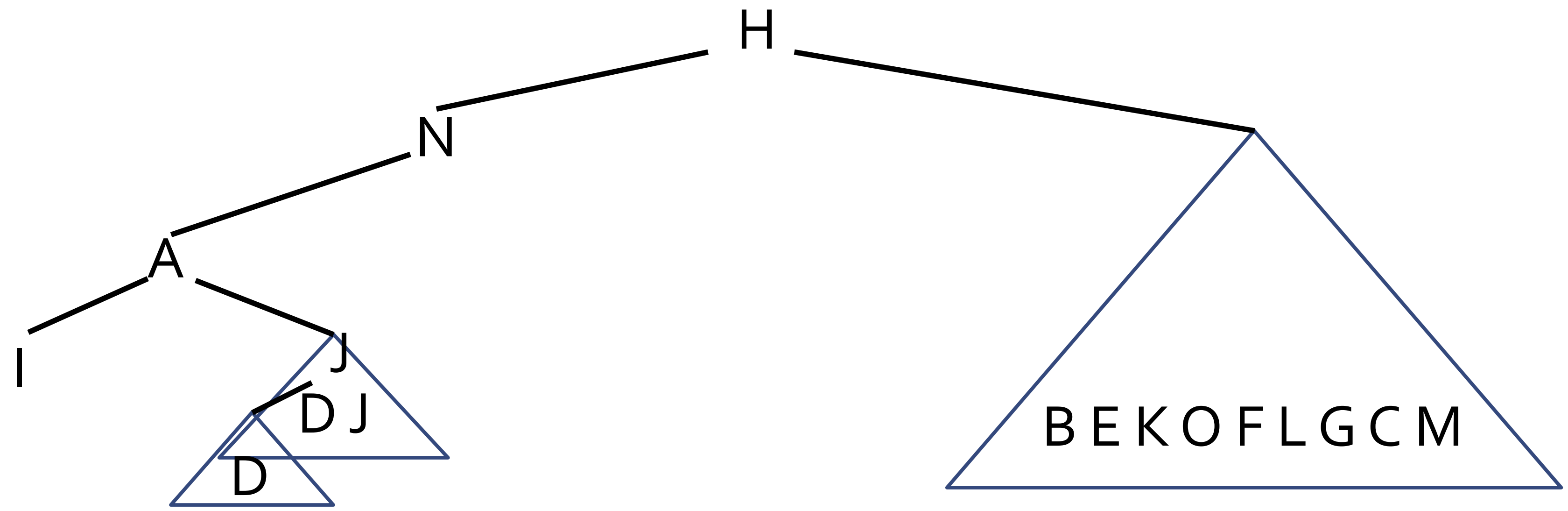
Pre-order: H N A I J D O B K E C L F G M



Q10. Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M

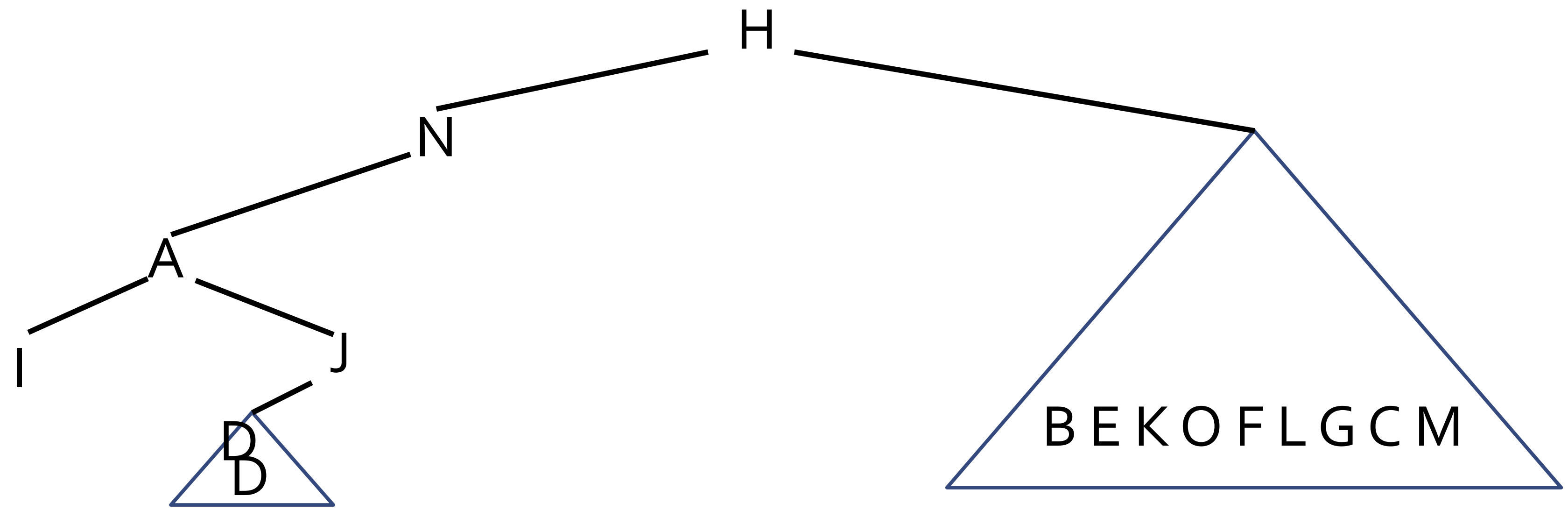


Q10.

Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

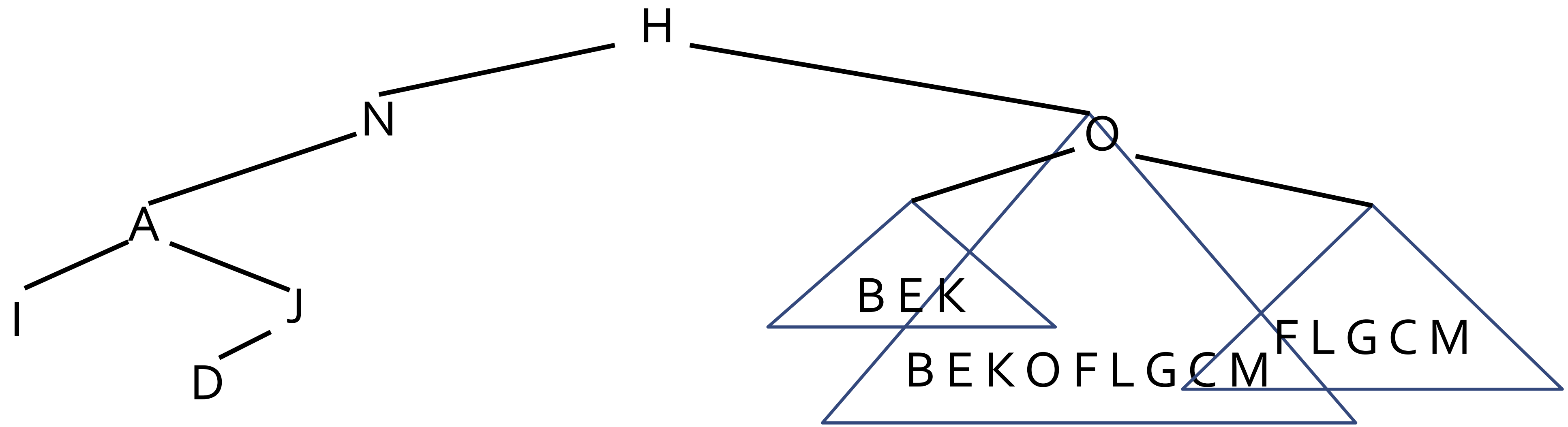
Pre-order: H N A I J D O B K E C L F G M



Q10. Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M

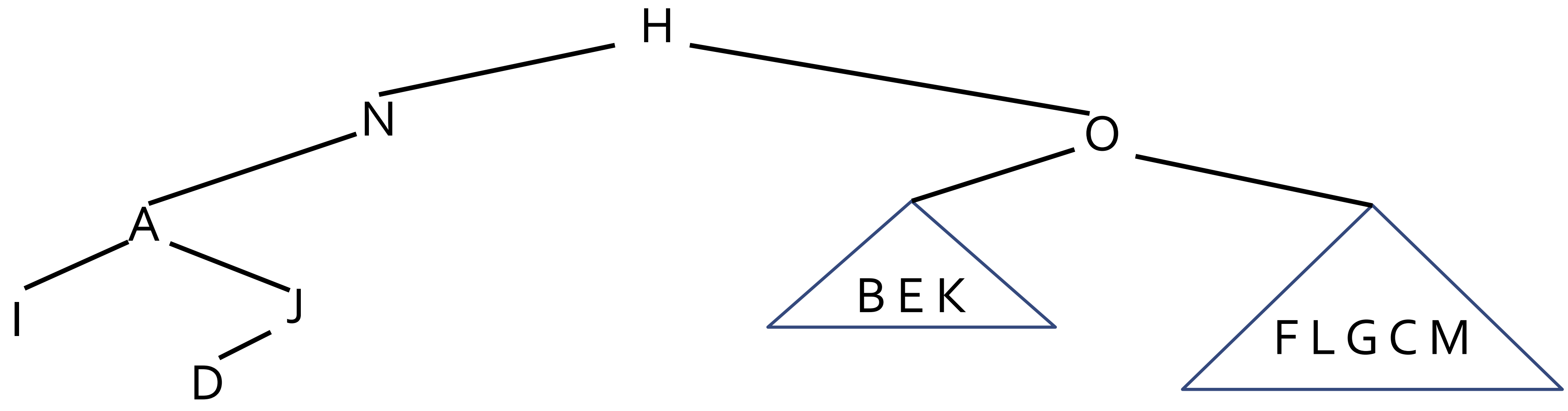


Q10.

Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M

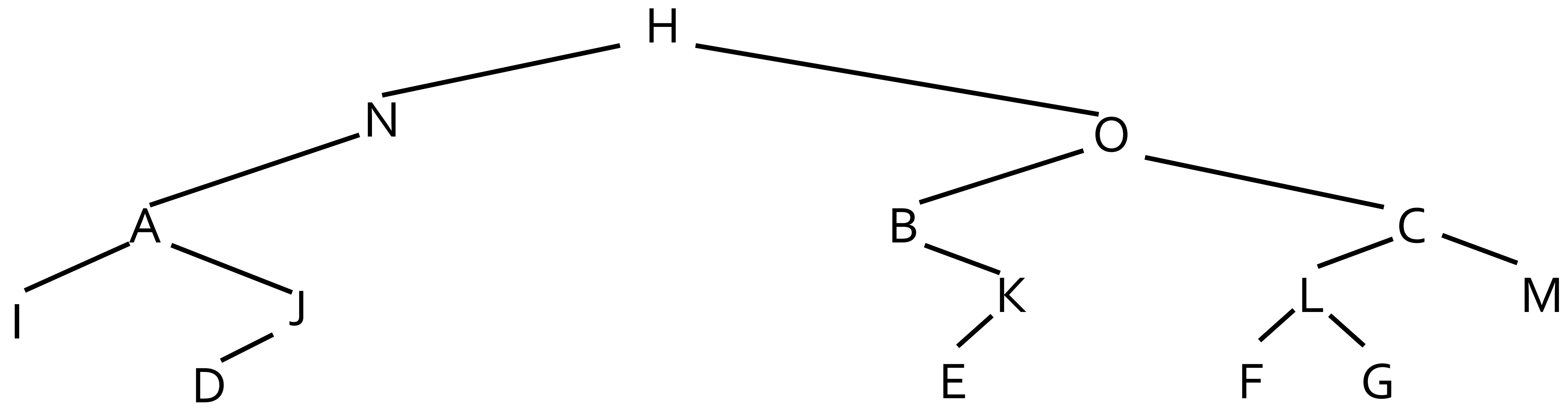


Repeat the process…

Q10. Construct the binary tree give the following in-order and pre-order traversal of the tree:

In-order: I A D J N H B E K O F L G C M

Pre-order: H N A I J D O B K E C L F G M



Final binary tree


You Can Do it

THE END