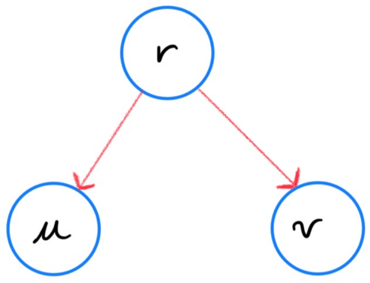
CS201

Assignment 2

Kushagra Srivastava

Question 1

The following graph fulfills the required conditions,



Question 2

In order to prove that R is an equivalence relation we need to show that it is *Reflexive, Symmetric* and *Transitive.*

- **Reflexive**
  For every vertex u in a graph, there is a path of zero length, to that same vertex. Thus, every vertex is reachable from itself. Hence, $\forall u \in V$, path uRu exists.
- **Symmetric**
  If for $u, v \in V$ the path uRv, exists then by definition of the relation itself, the path vRu also exists, hence the function is symmetric as well.
- **Transitive**

  Assertion: If uRv and vRw, exists for $u, v, w \in V$, then uRw, also exists.

  Proof.
  - The paths, uRv and vRw, do not intersect, i.e they have no common vertex other than v itself. In this case, the union of the paths, gives a path from u to w.
  - The paths, uRv and vRw, intersect, i.e they have at-least one common vertex say k_i other than v itself. In this case, we can just follow the path from vertex k_i to vertex w, and keep repeating the process, till no repeating vertex remains.

  From the above two cases we can conclude that R is a transitive relation.

Therefore, since R is **Reflexive, Symmetric and Transitive**, it is an Equivalence Relation.

Question 3

An Equivalence class C for a relation R is defined as $aRb \Leftrightarrow a, b \in C$.
Let V_1 and V_2 be the two equivalance classes, which by defintion are also strongly connected components.
Let $\{x_0, x_1, x_2, ..., x_n\}$ belong to V_1 and $\{y_0, y_1, y_2, ..., y_n\}$ belong to V_2.
Assuming there exists an edge between atleast one node each of V_1 and V_2.
Without Loss of Generality, consider this edge to be, $(x_0, y_0) \in E$.
There can not be an edge $(y_0, x_0) \in E$, since if this edge exists, it would imply,

$$x_0 R y_0 \Leftrightarrow y_0 R x_0$$

Assertion: There is no edge from V_2 to V_1

Proof.
Let's assume there exists an edge between y_i and x_j, i.e (y_i, x_j).
Since there is an edge (x_0, y_0) from V_1 to V_2, and there is always a path to x_0 from any $x_j \in V_1$, as it is strongly connected, and similarly there is always a path from y_0 to any $y_i \in V_2$ as it is also strongly connected. Therefore we can say that there is a path from any vertex x_j of V_1 to any vertex y_i of V_2.
Similarly, due to the existence of the edge (y_i, x_j), using the same argument as above, we can state that there is always a path from any vertex of V_2 to any vertex of V_1.
Thus, for any vertex $x_i \in V_1$ and $y_j \in V_2$, there is a path from x_i to y_j meaning that $x_i R y_j$ which is a contradiction since V_1 and V_2 are disjoint equivalence classes.
Therefore, there is no edge from V_2 to V_1.

Generalizing the above result, we can state that all the edges, if existing, are from V_1 to V_2 or V_2 to V_1.

Question 4

Claim: For a Directed Tree, the Root Node has an Indegree 0.

Proof.
Let's assume that no node has an indegree 0. This combined with the given definition in the problem, implies that the root has an indegree greater than 1, while the other $|V|-1$, vertices have indegree exactly 1.
Now, we can pick a node, and since its indegree is ≥ 1, this implies that we can go to its parent node. This parent node also has an indegree which is ≥ 1. Since, the tree is finite, this process can be repeated, until we reach a node which has already been visited. This implies that a cycle exists in the tree. Thus, giving a contradiction.
Therefore, the indegree of the root is 0.

1. For a directed tree, $|E| = |V| - 1$.

Proof.
Every edge in a tree will contribute to once to the outdegree of some vertex and the indegree of some vertex. This implies that the Sum of outdegree or indegree, will give us the number of edges.
Indegree $= (|V|-1) \times 1 + 1 \times 0$.
Thus,

$$|E| = |V| - 1$$

2. For any two vertices u and v, $u \neq v$, if there is a path from u to v then there is no path from v to u.

Proof. Let us assume that in a directed tree there exists, a path from u to v, and from v to u. Let the root node(i.e the node with indegree 0) be denoted by r. There are three possible cases.

- **Case 1:** r, lies on either one of these paths.
  WLOG, let's assume that r lies on the path from u to v. This implies that there exists a path (u, r) and a path (r, v). Thus the indgree of r, has to be ≥ 1, which is a contradiction to its definition itself.
- **Case 2:** r, lies on neither one of these paths.
  In a tree, there is always a path from the root to every node. This means, that apart from the path from v to u, there is a different path from r to a vertex which lies somewhere on the path (u, v), including the vertices u and v. This implies that the indegree of some vertex (not necessarily u or v, but on each of the paths (u, v) and (v, u) is greater than 1, which is a contradiction.
- **Case 3:** r, is equal to one of u and v.
  In this case, if there exists a path (u, v) and (v, u), the indegree of the root is not 0, which is a contradiction.

Therefore, neither of these cases are possible. Hence, for any two vertices u and v, $u \neq v$, if there is a path from u to v then there is no path from v to u.

Question 5

The number of binary trees that can be drawn on a plane sheet of paper, can be derived easily by using a recursive definition.
Let T_n be the number of binary trees with n nodes.
We have, $T_0 = 1$ and $T_1 = 1$.

For all $n \geq 1$, T_n satisfies the given recurrence,

$$T_n = \sum_{i=0}^{n-1} T_i.T_{n-i-1}$$

Proof.
A tree T with n nodes has a root with two sub-trees, say t_1 and t_2 as children (both of which are binary trees as proved below). Since the root of T is the root node,t_1 and t_2 must have $n-1$ nodes together (i.e. if t_1 has i nodes then t_2 has $n-i-1$ nodes).
The number of ways to make this tree for a given i, is $T_i.T_{n-i-1}$.
Summing this over all i's gives,

$$T_n = \sum_{i=0}^{n-1} T_i.T_{n-i-1}$$

Hence, we have established a recurrence relation to solve for the number of binary trees.

Claim:
The two graphs obtained on removing the root of a binary tree, are also binary trees.

Proof.
Consider one of these graphs.

- **Connectivity:**
  For two vertices a and b in the graph. Since they were originally a part of the directed binary tree T, they were connected. Consider this path, say $\{a, v_1, v_2...v_k, b\}$
  - **Case 1 :** This path has vertices from the other graph as well.
    Consider the first such vertex u, which has its predecessor in the first graph. The parent of u, will belong in the second graph. Thus, it will have a edge from within the second graph and an edge from our path. This implies that the indegree of $u \geq 2$. This is not possible
  - **Case 2 :** This path contains the root.
    This is clearly not possible, as it would imply that the root has a edge incident on it.

  Thus, the path only has vertices from its own graph in it only.
- **Indegree:**
  Consider the children c_1, and c_2 of the root, both of these had an indegree 1 each, due to the edge incident on them from the root. Now, since the root is removed this indegree becomes 0. The other vertices are not altered in any way, hence they still have an indegree of 1.

Thus these two statements prove that the resulting graphs are also binary trees.

Question 6

Let the Generating Function be represented by,

$$G(x) = \sum_{n\geq 0} T_n x^n = \sum_{n\geq 0}\sum_{i=0}^{n-1} T_i.T_{n-i-1}x^n$$

Since, $T_0 = T_1 = 1$,
This implies,

$$G(x) = 1 + \sum_{n\geq 1}\sum_{i=0}^{n-1} T_i.T_{n-i-1}x^n$$

Which, by a change of variables $z = n-1$ can be written as,

$$G(x) = 1 + x\sum_{z\geq 0}\sum_{i=0}^{z} T_i.T_{z-i}x^z$$

Using **Claim 1**, this is equivalent to,

$$G(x) = 1 + xG(x)^2$$

Solving this quadratic equation in $G(x)$ we get

$$G(x) = \frac{1 \pm \sqrt{1-4x}}{2x}$$

Focusing on $\sqrt{1-4x}$ now, the binomial expansion using binomial theorem is:

$$\sqrt{1-4x} = \sum_{i=0}^{\infty}\binom{\frac{1}{2}}{i}(-4)^i \cdot (x)^i$$

Using **Claim 2**,

$$\sqrt{1-4x} = 1 - \sum_{i\geq 1}\frac{2}{i}\binom{2(i-1)}{i-1} * (x)^i$$

We have,

$$1 - \sqrt{1-4x} = \sum_{i\geq 1}\frac{2}{i}\binom{2(i-1)}{i-1} * (x)^i$$

Therefore,

$$G(x) = \frac{\sum_{i\geq 1}\frac{2}{i}\binom{2(i-1)}{i-1}\cdot(x)^i}{2x} = \sum_{i\geq 1}\frac{1}{i}\binom{2(i-1)}{i-1}\cdot(x)^{i-1} = \sum_{i\geq 0}\frac{1}{i+1}\binom{2i}{i}\cdot(x)^i$$

Thus the coefficient for x^n, can be written as,

$$\frac{1}{n+1}\binom{2n}{n}\cdot(x)^n$$

Claim 1:

$$\sum_{n\geq 0}\sum_{i=0}^{n} T_i.T_{n-i}x^n = \left(\sum_{n\geq 0}^{z} T_n \cdot x^n\right)^2$$

Proof.
Let, the generating function $A(x)$,

$$A(x) = \sum_{n\geq 0} T_n \cdot x^n$$

Expressing the Left Hand Side of the equation in terms of this generating function as,

$$\sum_{n\geq 0}\sum_{i=0}^{n} T_i.T_{n-i}x^n = \sum_{n\geq 0}\left(\sum_{i=0}^{n} T_i.T_{n-i}\right)x^n$$

From the Convolution property of Generating Functions, this becomes,

$$\sum_{n\geq 0}\left(\sum_{i=0}^{n} T_i.T_{n-i}\right)x^n = A(x)\cdot A(x)$$

Thus, we have,

$$\sum_{n\geq 0}\sum_{i=0}^{n} T_i.T_{n-i}x^n = \left(\sum_{n\geq 0}^{z} T_n \cdot x^n\right)^2$$

Claim 2: For $i > 0$

$$(-4)^i\binom{\frac{1}{2}}{i} = -\frac{2}{i}\binom{2i-2}{i-1}$$

Proof.
Simplifying the LHS, yields,

$$(-4)^i\binom{\frac{1}{2}}{i} = (-4)^i\frac{\frac{1}{2}\cdot\left(\frac{1}{2}-1\right)\cdot\left(\frac{1}{2}-2\right)\cdot\ldots\cdot\left(\frac{1}{2}-i+1\right)}{i!}$$

Which gives,

$$(-4)^i\binom{\frac{1}{2}}{i} = (-4)^i(-1)^{i-1}\cdot\frac{1\cdot 3\cdot 5\cdot\ldots\cdot(2i-3)}{2^i\cdot i!}$$

This becomes,

$$(-4)^i\binom{\frac{1}{2}}{i} = -(2)^i\cdot\frac{1\cdot 3\cdot 5\cdot\ldots\cdot(2i-3)}{i!} = -(2)^i\cdot(i-1)!\cdot\frac{1\cdot 3\cdot 5\cdot\ldots\cdot(2i-3)}{i!\cdot(i-1)!}$$

On Further Simplification,

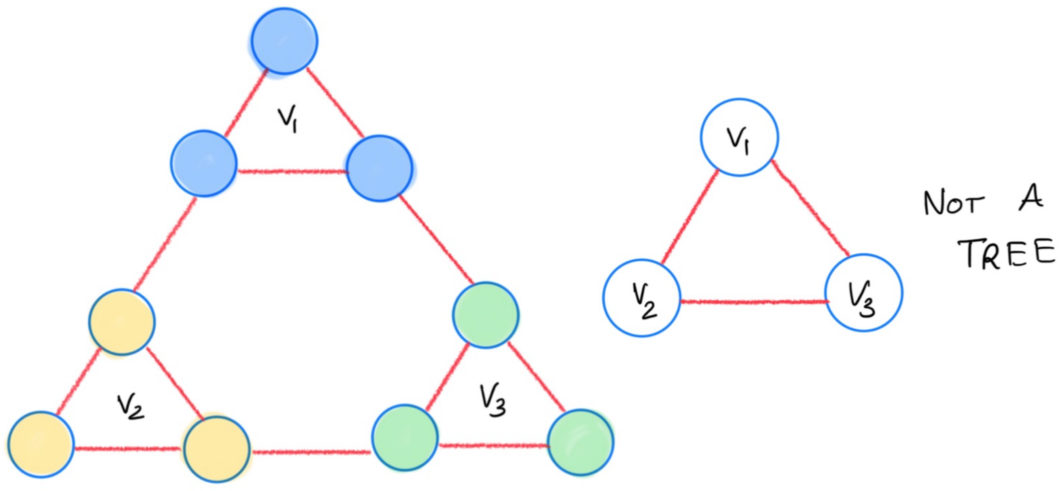
$$-2\cdot 4\cdot 6\cdot 8\ldots(2i-2)\cdot\frac{1\cdot 3\cdot 5\cdot\ldots\cdot(2i-3)}{i!\cdot(i-1)!}$$

Which can be written as,

$$(-4)^i\binom{\frac{1}{2}}{i} = -2\cdot\frac{(2i-2)!}{i!\cdot(i-1)!} = -\frac{2}{i}\binom{2i-2}{i-1}$$

Question 7

The following graph serves as a counter-examlple to the above statement.



This cycle is not a tree, as there exists non-unique paths between any two vertices, which is not allowed for trees in a directed graph.

Question 8

Claim 1. An undirected graph, which is a tree is connected and acyclic.

Proof.
Let, G, be an undirected tree.
As discussed in the lectures, there exists a unique path between any two vertices, $u, v \in G$. Thus, G, by definition is connected.
Let us assume that G has a cycle, with vertices $v_1, v_2, v_3 ... v_k, v_1$.
Consider two vertices v_i, v_j, with $i < j$ in this cycle. We have a path from u to v, in both the direction.

- Path from v_i to v_j : This is just the contiguous path in the cycle with elements v_r, with $i \leq r \leq j$.
- Path from v_j to v_i : This contains three paths, the path from v_i to v_k, from v_k to v_1 and from v_1 to v_j.

Moreover, since it is an undirected graph, the reverse of both of these paths is also a valid path. This implies that there are two paths form v_i to v_j, which violates the argument that there must be a unique path in a tree.
Thus we have achieved a contradiction. There can be no cycles in a tree.

Claim 2. An undirected graph, which is connected and acyclic, is a tree.

Proof.
Let, G, be an undirected graph, which is connected and does not have any cycles.
Therefore, a path exists between any two vertices.
Let us assume that there exists more than one unique path between two vertices u, v.Name them as Path-1 and Path-2.
Path-1(u, v) and, since it is an undirected graph, the reverese of Path-2, as Path-2'(v, u). A combination of these two paths, without the common vertices, yields a cycle in G, which contradicts our assumption. Thus, there exists only one unique path between two vertices, Thus G, is a tree

Question 9

Let there exist a cycle $v_1, v_2, v_3 ... v_k, v_1$, in a directed graph G.
Consider two vertices v_i, v_j, with $i < j$ in this cycle. For them to lie in the same strongly connected components, we need to show that there exists a path from u to v, in both the direction.

- Path from v_i to v_j : This is just the contiguous path in the cycle with elements v_r, with $i \le r \le j$.
- Path from v_j to v_i : This contains three paths, the path from v_i to v_k, from v_k to v_1 and from v_1 to v_j.

Hence, every pair of vertices in the cycle have paths in both directions between them. Therefore, they lie in the same connected component.

Question 10

Claim 1. If the undirected graph has a Spanning Tree, then it is Connected.

Proof.
According to the given defintion, a spanning tree is a subgraph that includes all the vertices of the original graph and is a tree, which means it is a *connected acyclic graph.*
Given, G is an undirected graph with a spanning tree T.
Since all the vertices of G are in T, this means that there is a path in T between any pair of vertices in G. Since T is a subgraph of G, this path also exists in G.
Therefore, G must be connected.

Claim 2. If the undirected graph is connected, it has a Spanning Tree.

Proof.
We can construct a spanning tree by using an algorithm such as, depth-first search (DFS).
Depth First Search

1. Start at a arbitary vertex as the root node.
2. Mark the current node as visited to keep track of nodes you have explored.
3. Explore the unvisited neighbors of the current node. You can choose a neighbor to visit first; there's no strict order.
4. If you find an unvisited neighbor, repeat steps 2 and 3 for that neighbor.This process of visiting and marking nodes continues recursively until there are no unvisited neighbors.
5. When there are no more unvisited neighbors, backtrack to the previous node. This is done by returning to the previous recursive call in the algorithm.
6. Repeat steps 3 to 5 for any remaining unvisited neighbors of the current node until all nodes have been visited.

Start with an arbitrary vertex in the connected graph G. Perform a DFS from this vertex, marking the visited vertices. The result will be a tree that includes all the vertices and edges reachable from the starting vertex.
Since G is connected, the DFS will visit all vertices. The tree obtained from the DFS is a connected acyclic subgraph, which is the definition of a spanning tree.
Therefore, if a graph is connected, it is always possible to construct a spanning tree for it.

In conclusion, we have shown both directions of the statement: an undirected graph has a spanning tree if and only if it is connected.

Question 11

Claim 1. The Minimum Weight Subgraph is a Spanning Tree.

Proof.
The Minimum Weight Subgraph, denoted by G_{MWS}, is a connected subgraph, thus, it spans all vertices in G, and is acyclic as well, as removing the cycle will reduce its weight.
Also, removing any edge from G_{MWS}, will result the graph in being disconnected.
Therefore, G_{MWS} is acylic, connected and covers all vertices of G, thus it is a Spanning Tree.

Question 12

Let T be the spanning tree found by using the given algorithm, and T_{MST} be the Minimum Spanning Tree of the given graph.

Assertion: $T = T_{MST}$

Proof.
Assume $T \neq T_{MST}$.
Therefore, $T - T_{MST} \neq \phi$
Let (u, v) be any edge in $T - T_{MST}$.
When (u, v) was added to T, it was the edge with the least edge such that $u \in U$ and $v \notin U$ was satisfied.
Since T_{MST} is a Minimum Spanning Tree, there must be a path from u to v in T_{MST}.
This path begins in U, and ends in somewhere not in U, so there must be some edge (x, y) along that path where $x \in U$ and $y \notin U$.
Since (u, v) is the minimum weighted edge possible , we have $c(u, v) < c(x, y)$.
Let $T'_{MST} = T_{MST} \cup \{(u, v)\} - \{(x, y)\}$. Since (x, y) is on the cycle formed by adding (u, v), this means T'_{MST} is a spanning tree.
However,

$$c(T'_{MST}) = c(T_{MST}) + c(u, v) - c(x, y) < c(T'_{MST})$$

This contradicts the fact that T_{MST} is the Minimum Spanning Tree.
We have reached a contradiction, so our assumption must have been wrong.

Question 13

The Weight of a Minimum Spanning Tree of the Graph is at most the Minimum Length Tour.

$$z_{MST} \leq z_{MLT}$$

Proof.
Let the H be the Minimum Length Tour and it has a weight of z_{MLT}.
Removing one edge from the Minimum Length Tour, breaks the cyclic nature of the walk, and yields a Spanning Tree, with cost z_{ST}, which obviously is greater than the weight of the Minimum Spanning Tree, z_{MST}

$$z_{MST} \leq z_{ST} \leq z_{MLT}$$

Thus,

$$z_{MST} \leq z_{MLT}$$

Triangle Inequality.
For the Travelling Salesman Problem, in any graph if we have a path (u, w) and (w, v), then the length or cost of the path, will always satisfy the following,

$$c_{(u,w)} + c_{(w,v)} \leq c_{(u,v)}$$

Question 14

The Weight of the Minimum Length Tour is at most twice the Weight of a Minimum Spanning Tree.

$$z_{MLT} \leq 2z_{MST}$$

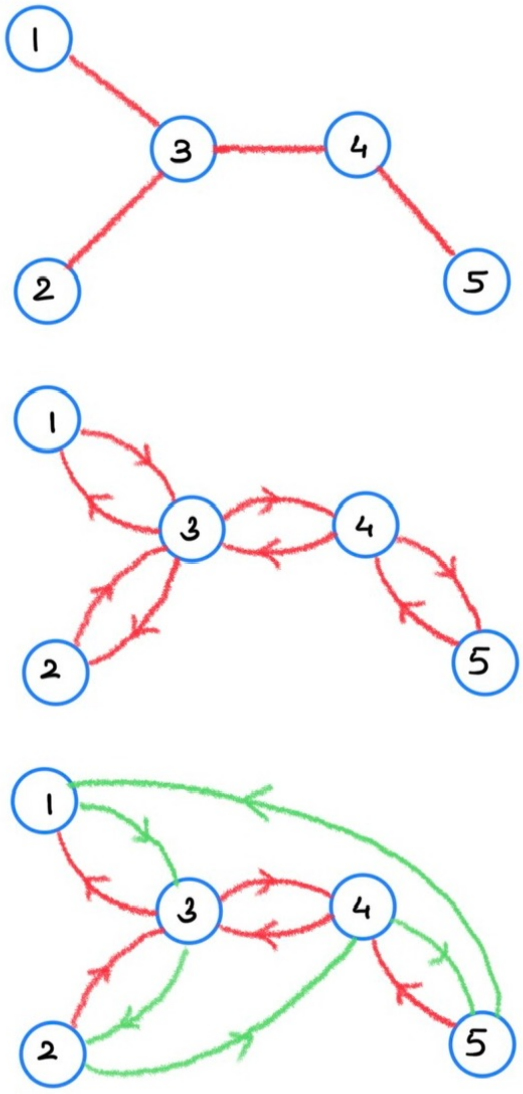
Proof.
We have a Minimum Spanning Tree, T_{MST}, with a weight z_{MST}.
Replace each Edge in the tree by two copies of itself (along with directions, one in each direction), as show in the figure. Now, this resulting graph, say T_E is Eulerian. (A graph is said to be Eulerian if there exists a tour that visits every edge exactly once). Since, every edge is included twice,

$$z_{MLT} \leq z_E \leq 2z_{MST}$$

Thus,

$$z_{MLT} \leq 2z_{MST}$$

In order to obtain the Minimum Length Tour, which has a length which is at most equal to the Eulerian Tour, we remove all but the first occurrence of each node in the sequence using the **Triangle Inequality**.



Alternate Solution to Question 14

The Weight of the Minimum Length Tour is at most twice the Weight of a Minimum Spanning Tree.

$$z_{MLT} \leq 2z_{MST}$$

Proof.
Performing a DFS Traversal on the graph, according to the alogrithm mentioned in the above problem, we will be traversing every vertex twice at max, once while traversing downward from the root and the other while traversing back recursively. Thus,

$$z_{MLT} \leq z_{Tour} \leq 2z_{MST}$$

Thus,

$$z_{MLT} \leq 2z_{MST}$$