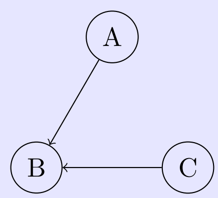
CS 201 Assignment 2

Mahaarajan J

31 October 2023

1.

One example of a directed graph could be a graph G with
3 vertices:A,B,C and
directed edges: (A,B) and (C,B).



Clearly in this directed graph there is no path between (A,C) or (C,A) as the only edge from either A or C ends up at B from where there are no edges (outdegree(B)=0).
But if the graph was undirected (if we ignore the edge directions) then due to edges (A,B) and (B,C) existing we can have the path A $\longrightarrow$ B $\longrightarrow$ C between A and C and vice versa.

Hence we have found a valid example graph G.

2.

We shall prove that relation R as defined to be an equivalence relation by proving R to be reflexive,symmetric and transitive:

Reflexive: The relation R is reflexive as there will be a path from any vertex u to itself (the path will only have a single vertex u) hence (x,x) ϵ R for all x in V (here u=v=u_0)

Symmetric: from definition of R (u,v) ϵ R if there is a path from u to v and vice versa. Hence (v,u) will also ϵ R for every (u,v) ϵ R as there is a path from v to u and vice versa (path (v,u) exists as (u,v) is a member of R).

Transitive: If (u,v) and (v,w) ϵ R i.e (u,x_1,x_2...v) and (v,$y_1, y_2, y_3 ... w$) are paths then for any u,v,w in V then there will exist a path from u to w through v (u,x_1,x_2...v,$y_1, y_2, y_3 ... w$) (this is a valid path as all edges in the path formed by adjacent vertices are already in the graph due to the existance of the previous 2 paths) and vice versa. Hence (u,w) will belong to R

Hence Proved

3.

We shall try to prove this statement by method of contradiction.
The contradiction would be:
"There exists a,c ϵV_1 and b ϵV_2 such that (a,b) and (b,c) are edges or vice versa (interchange V_1 and V_2".
(we have taken the edges in opposite directions and clearly, the contradiction of this statement would be that, there is no such (a,b,c) which would mean that all edges would be in the same direction)

Now there is a path from a to b (due to the direct edge) and there is a path from b to a through c (as there is an edge between b and c, c and a are connected). Therefore (b,a) ϵ R. But a and b are from different equivalence classes. Hence contradiction. and distinct equivalence classes must be disjoint.

Proof:
Let A,B be 2 equivalence classes of S ,R be an equivalence relation on S and let a ϵ A and b ϵ B.
If aRb then xRb for all x ϵ A as x,a belong to the same equivalence class and R is transitive. Hence b ϵ A as well.
Also if aRb we have yRa for all y ϵ B (as y,b in same equivalence class and R is transitive) hence by previous argument y ϵ A as well.
Hence if aRb A,B become the same equivalence class. Therefore distinct equivalence classes must be disjoint.
Hence Proved

Hence we have arrived at a contradiction
Therefore edges from V_1 to V_2 are in the same direction,i.e.,either all are from vertices in V_1 to vertices in V_2 ,or all are from vertices in V_2 to vertices in V_1.
Hence Proved

4.

Since in a directed tree each vertex apart from root (claim: V whose indegree=0) has indegree=1 and total number of edges would be sum of indegrees of each vertex.
As we will be counting each edge exactly once as for every edge (a,b) we will count it exactly once in the indegree of b, Hence by adding the indegrees of all vertices we will get the total number of edges.

Therefore the total number of edges will be number of vertices-1 in a directed tree.
Hence Proved

Claim: There exists a vertex with indegree=0 called root in a directed tree (every other vertex has indegree=1). We will prove this claim by induction.

Base case: For n=1 we will have only one vertex and its indegree=0. hence valid.

Induction hypothesis: Let us say this is true for n vertices and V be the root then let us add another vertex U to this tree.
Case 1: There is no edge between root and U. Then we add U to any other vertex of the graph say X as (X,U) as the indegree of X is already 1 and it can't increase. Then we will have a tree of size n+1 with all vertices (including U) indegree=1 except V whose indegree remains 0.
Case 2: We add it to the root as (V,U). Then again we will have a tree of size n+1 with all vertices (including U) indegree=1 except V whose indegree remains 0.
Case 3: We add it to the root as (U,V). Then we will have indegree of U=0 and it will be the new root as all other vertices including V have indegree=1.

Hence proved the claim

To Prove: If there is a path from u to v then there is no path from v to u.
We shall try to prove this by contradiction.
Let there be a path from u to v then we have created a cycle from u to u $(u, x_1, .v, ..u)$ (all x_i are unique) since there is a path from u to v and v to u.

Claim: No cycles can exist in a directed tree.

Proof:Because of the cycle all vertices must have their indegerees at least 1 (as one edge from the cycle will come into the vertex). Since it is a directed graph the only possibility is that all of them have indegerees=1. Now we have 2 cases
Case 1: The root is part of this cycle.
Then it is a straightaway contradiction as the indegree of the root must be 0.
Case 2: The root is not a part of the cycle. Then since indegerees can't be more than 1 there can be no incoming edges to any element of the cycle (apart from the edges of the cycle). Since the directed tree is connected and there can be no incoming edges from outside to any vertex in the cycle, the outside vertices can only be connected by edges from the cycle to that vertex (say V), making the indegree of V atleast 1. Hence V can't be the root as indegree of root is 0.
Therefore the root must be a part of the cycle (as we have proved it can't be an outside vertex).
Hence both cases end in contradiction,
therefore no cycles can exist in a directed tree.

Hence Proved the claim

Therefore given there is a path from u to v there can't be a path from v to u.

Hence Proved

5.

In order to derive the recursion for number of binary trees let us first select one node as the root. Let the number of trees possible for n nodes be **t(n)** Now let us say the left-subtree has i nodes where i is a number greater than or equal to 0. Then the number of nodes in the right subtree would be n-i-1 (as total number of nodes is n).
Hence the number of trees for this configuration where there are i nodes to the left would be **t(i)t(n-i-1)** (As there are t(i) possibilities for the left subtree and t(n-i-1) possibilities for the right and the 2 are independent of each other)
Now i can be any number ranging from 0 to n-1. Hence total number of trees **t(n)** would be $\sum_{i=0}^{n-1}$**t(i)t(n-i-1)**
Base cases: t(1)=1,t(0)=1(empty tree only 1 possibility).

Hence found a recurrance relation

Claim: the two graphs (left and right subtrees) obtained by removing the root of a binary tree are also binary trees.
Proof: We will have to prove connectivity and indegrees to prove it is a tree by q4.

Connectivity: Let us take 2 vertices x,y (not the root as the root is not a part of the sub graph anyway) in the original graph, as the original graph was a tree there will be a path between them say $(x, v_1, v_2...y)$ now there are 3 cases:

Case 1: The path has vertices from both the subgraphs.
Let us consider the first instance of such a vertex u in the 2nd graph whose predecessor in the path is in the 1st graph say v. Now as u is in the 2nd graph its parent will be in the 2nd graph say p. Hence there will be edges (v,u) and (p,u) which would make the indegree of u atleast 2. This is not possible as u was a vertex in the previous tree and by definition indegree of u is atmost 1 (and we are not adding any new edges while making the subtrees).
Contradiction

Case 2: The path between the vertices contained r the root of the original graph which we removed. The vertex x is non root, hence if the root is part of the path there has to be a vertex j such that (j,r) is an edge which would make the indegeree of r non zero. This is a contraction as the orginal graph is a tree and its root must have indegree=0.
Contradiction
Thus the path has vertices from its subgraph alone. Hence the subgraphs after removal of the root will still be connected.

Indegeree: The children of the root c and d will now have indegree=0 as we have removed the edges (r,c) and (r,d). Other vertices have not been altered in any way hence their indegree will continue to be 1.
Hence the subtrees will have c,d as roots respectively and all other vertices will have their indegree as 1.

Max number of children: This will still remain 2 as we are not adding any edge or vertex to the existing binary tree
Thus the 2 subgraphs formed are also binary trees.
Hence Proved

6.

Solving the recurrance relation using Generating functions.
We first rewrite the recurrance as t(n+1)=$\sum_{i=0}^{n}$t(i)t(n-i) and t(0)=t(1)=1.
Let us take the generating function G(x)=$\sum_{i=0}^{\infty} t(i)x^i$ with the required numbers as the coeffecients. Now let us square this function $G^2(x)$ now the coefficient of x^i in this sequence would be $\sum_{j=0}^{i} t(j)t(i-j)$ (we will collect all those exponents that add up to i from both the sequences of the product) and this summation from the recursive relation would be t(i+1).
Thus $G^2(x)$ will be $\sum_{i=0}^{\infty} t(i+1)x^i$ which would simply be $(G(x)-1)/x$ (remove the constant term and divide by x to get $t(i)x^{i-1}$ from 1 to inf which would simplify to $t(i+1)x^i$ from 0 to inf). Hence we have the quadratic equation in G(x)

$$xG^2(x) - G(x) + 1 = 0$$

As it is a quadratic equation on applying the quadratic formula and choosing the root such that G(0)=1 we get.

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

Now the nth term of this sequence would be our required answer for t(n).

Now solving for the nth term we have. Expanding $\sqrt{1-4x}$ now, the binomial expansion using binomial theorem is:

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

Claim: For $i > 0$ we have

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

Proof:Starting with LHS:

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

Simplifying the binomial coefficient we have:

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

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

Multiply and Divide by $(i-1)!$ after substituting in lhs and simplification gives

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

This becomes,

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

Hence,

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

which is exactly the expression in RHS
Hence proved.
Therefore,

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

This implies,

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

Now since $G(0)$ is defined ($G(0) = T_0 = 1$), as discussed before we take the root with negative sign as the one with positive sign is not defined at 0. Therefore,

$$G(x) = \frac{\sum_{i=1}^{\infty} \frac{2}{i}\binom{2i-2}{i-1}(x)^i}{2x}$$

Replacing i with $i+1$ this simplies to,

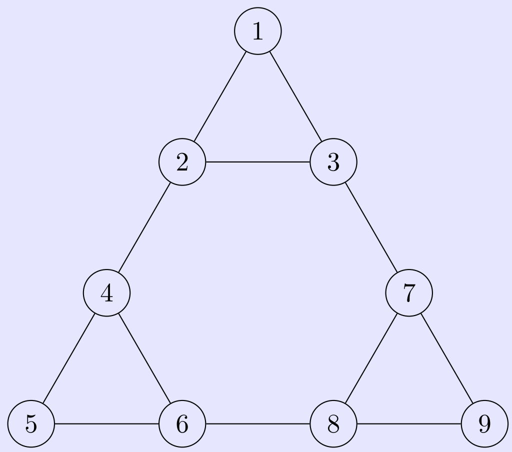
$$G(x) = \sum_{i=0}^{\infty} \frac{1}{i+1}\binom{2i}{i}(x)^i$$

Now the coefficents of x^n in G(x) are the terms t(n) that we need Hence :

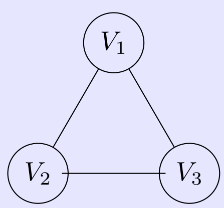
$$t(n) = \frac{1}{n+1}\binom{2n}{n}$$

7.

The given statement is wrong as we can find a counter example. The counter-example is the graph G having 9 vertices:



Now this graph has 3 connected components V_1=(1,2,3),V_2=(4,5,6) and V_3=(7,8,9) and the graph H will have the vertices V_1, V_2, V_3 and edges $(V_1, V_2), (V_2, V_3), (V_3, V_1)$.



Clearly H can't be a tree as there is a cycle in H (V_1, V_2, V_3, V_1) and a tree can't have a cycle because by definition of undirected tree there has to be a unique path between any 2 vertices. Here if we take the vertices (V_1, V_3) we will have 2 paths (V_1, V_3) and (V_1, V_2, V_3). Thus the given graph H will not be a tree

Hence disproved

8.

Basically we need to prove that undirected graph is tree $\longleftrightarrow$ connected and no cycles. We consider the definition of tree equivalent to there is a unique path between any 2 vertices.

1) $\longrightarrow$
As the graph is a tree there is a path between any 2 vertices hence the graph is connected. Now let us consider that the graph has a cycle say $(x, v_1, v_2, ..v_k, x)$. Let us consider paths between the vertices x and v_1 and x. One is the obvious (x,v_1) and the other is $(x, v_k, ..., v_2, v_1)$ (we are allowed to reverse the order as the graph is undirected). Hence we have arrived at a contradiction. Therefore there is no cycle in the graph
Hence Proved

2)$\longleftarrow$
As the graph is connected there is a path between any 2 vertices. Let us consider the existence of 2 paths between x and y (we are proving by contradiction) $(x, v_1, v_2, ...y)$ and $(x, u_1, u_2, ...y)$. Now we will claim that such a graph with these 2 paths will have a cycle.
Let v_k be the first vertex at which v_i is not equal to u_i such a v_k must exist as the paths are distinct and let u_l be the first vertex after k at which some v_i $(i, l > k)$ is equal to u_l (let us consider y as u_m or v_n). Now such a u_l will exists as both the paths end at the same vertex so u_l can be y if there is no repition in between. Now the path (let x be v_0) $(v_{k-1}, v_k, ...v_i, u_{l-1}, ...u_k)$ (as v_i is equal to u_l) also by definition u_k is equal to v_k hence the given path is a cycle and our graph should not have any cycle.
hence we have arrived at a contradiction.

Hence Proved

9.

Claim: In a cycle any 2 points have a path between them.
Proof: Let us consider the cycle $(v_1, v_2, ...v_n)$ such that $v_1 = v_n$. Now let us consider 2 vertices v_i and v_j. If $i > j$ then the path from v_i to v_j would be $v_i, ..v_n, v_2, ..v_j$. If $i < j$ there is trivially a path $v_i, ..v_j$ from the sequence of the cycle. Hence there is always a path from v_i to v_j where they are any 2 points of a cycle. Hence by interchanging j and i we get that a path exists from v_j to v_i as well. Hence we can say that uRv for all u,v in the cycle.
Hence Proved

Hence as uRv they must be a part of the same equivalence class (strongly connected component) as distinct equivalence classes are disjoint (refer q3 for this proof).
Therefore in a directed graph all vertices in a cycle lie in the same strongly connected component.

Hence Proved

10.

Spanning Tree $\longleftrightarrow$ undirected connected graph

1) $\longrightarrow$

By definition of spanning tree it contains all the vertices of the graph G. Since it is a tree all its vertices must be connected as in a tree we have a unique path between any 2 vertices. Therefore even if some extra edges are added to make the graph G the connectedness would still hold

Hence Proved

2) $\longleftarrow$

Let us take 2 cases:

Case 1: The given graph G does not have any cycles, then from q8 trivially the graph G itself is the spanning tree (as it is connected and has no cycles), E'=E. (G'=G)

Case 2: The given graph has cycles.

Claim: Any connected graph G has a subgraph G' such that G' is connected and does not have any cycles.

Proof:

Let us take the graph G with one cycle $(x, v_1, v_2, ..., v_n)$ and $v_n = x$. Now let us remove an arbitrary edge of the cycle (v_i, v_{i+1}) then this cycle will be broken i.e this sequence of vertices will no longer be a path (other cycles may exist). Now the graph will still be connected after the removal as the existence of the cycle before removal guarantees a path $(v_{i+1}, ...v_n, ..., v_i)$ between v_i and v_{i+1} after the removal of the edge. Hence all paths which initially had the edge (v_i, v_{i+1}) will now have the above sequence of vertices.

Hence there will still be a path between any 2 vertices and the subgraph will still be connected.

In a similar way we identify all the cycles and remove one arbitrary edge as discussed above till we have no cycles left to obtain a subgraph G' which is connected and does not have any cycles.

Hence Proved

Therefore we have found a subgraph G'=(V,E') of G=(V,E) such that G' doesn't have any cycles and by q8 it will be a tree.

Hence Proved

11.

To Prove: Min weight subgraph is a minimum spanning tree of the graph.

Claim: Minimum weight subgraph does not have any cycles.
Let a minimum subgraph G have a cycle. Then by the previous question it is possible to remove an arbitrary edge from the cycle whilst preserving the connectedness of the graph. Hence we get a subgraph of G' say G" with one less edge. Since weights of edges are positive (implicitly assumed as lengths of water pipes) removing an edge reduces the weight of the graph. Hence W(G") less than or equal to W(G') which is a contradiction as G' is the minimum weight subgraph.
Hence Proved

Let G' be the minimum wt subgraph. By the claim and previous question we can say that G' is a spanning tree. Let us consider other spanning trees F',H' etc. of them G' will be the minimum as from definition G' is the minimum weight subgraph (spanning trees are also subgraphs). Therefore since the minimum weight subgraph is a spanning tree it will be a minimum spanning tree.

Hence Proved

12.

The above algorithm to calculate the minimum spanning tree is called the Prim's algorithm and it works. Here is its proof that the spanning tree created by the given algorithm is the minimum spanning tree:

Let us consider T to be the spanning tree created by the algorithm and T' to be the minimum spanning tree. We will prove T=T' by contradiction.

Let T be different from T' and T-T' be null then:
As number of edges in a spanning tree would be $|V| - 1$ (from Klass theorm as discussed in class) in both cases and T-T' being null as that would imply T is a subgraph of T' which would only be possible if all their edges were same (as cardinality of edges is same) which will be a contradiction to the fact that T' is different from T

Hence if T is differnt from T, T-T' will not be null. Hence let us choose an edge (u,v) such that (u,v) is in T but not in T'.
When (u,v) was added to T it was the least cost edge crossing a cut (U,V-U) where U is the set of vertices containing u and not containing v (similar to the set U defined in the question).

Now as T' is MST there will be an edge (a,b) across (U,V-U) (due to connectedness there must be an edge between these 2 disjoint sets) such that weight of (u,v) is less than (a,b) (As (u,v) was added as opposed to (a,b) in our algorithm).

Now since (a,b) is on the cycle formed by adding (u,v) we can say that if we replace (a,b) with (u,v) the tree will still be spanning. Let us call this new tree as T".
w(T")=w(T')+w((u,v))-w((a,b)). Hence as w(u,v) is less than w(a,b) the cost of T" will be less than T'. Hence we have reached a contradiction to the fact that T' is MST.

Hence our initial assumption that T is different from T' is false therefore, T=T'.
Therefore the tree from the algorithm is an MST

Hence Proved

13.

To Prove: Length of minimum spanning tree is atmost the minimum length tour.

Let the weight of minimum length tour be W. Let us eliminate all the cycles (like in q10 by removing an arbitrary edge in the cycle) from the minimum length tour to get a spanning tree of weight W'.
Since we are removing edges and weights (lengths of roads) are positive we have W greater than or equal to W'. And since W' is a spanning tree by definition of MST the weight of W' will be less than the weight of MST. Hence the weight of MST will be atmost weight of MLT, and equality holds when there is just one vertex as both the lengths of MST and MLT will be 1.

Hence Proved

14.

To Prove: $W(T*) < 2 * W(T)$ where T* is the minimum length tour and T is the minimum spanning tree.
For this we will prove that there exists a path P through every vertex, returning to the original vertex such that W(P) is atmost 2*W(T).
Now W(T*) will be less than or equal to W(P) as by definition minimum length tour has the minimum weight out of all possibilities of P.

We can find a path P by applying the DFT algorithm (depth first traversal) on the MST. It's procedure starting from a given vertex u is
1) Visit a neighbour of u say v that has not already been visited, mark v as visited. If all the neighbours are visited then return to the last visited vertex using the same edge that was traversed to come to u (the current vertex)
2) Do DFT on v
3) go to step 1.

Now we shall prove that DFT visits all vertices and each edge is traversed exactly twice in the algorithm.

Visiting: Let us assume there is a vertex a which is reachable from the initial vertex u (there is a path between a and u) but it doesn't get visited. It is possible to choose such a vertex a with a neighbour b which is visited as the DFT always visits the 1st vertex v, its neighbours and so-on.
Now let us consider the vertex b which is visited. When the vertex b is visited from the algorithm all the unvisited neighbours of b will be visited including a. Hence we arrive at a contradiction as a will now be visited.
Hence Proved

Edge exactly twice: From the algorithm as we are visiting all vertices once we will be traversing each edge atmost once. Additionally when the neighbours of a particular vertex are exhausted we return to its parent using the same edge that was used to visit the vertex, so every edge is hence visited twice.
Now the algorithm does not go back to the current vertex again as it has been marked visited. Therefore the edges are never visited again.

Hence we have concluded that every edge is visited exactly twice, therefore W(P)=2*W(T) where T is the MST and since W(T*) is less than or equal to W(P) we have proved that $W(T*) <= W(T)$.

Hence Proved