CS201 Assignment 2: The Concept of Graphs

Dhruv Gupta

October 2023

Solution 1

Consider the graph $G = (V, E)$, with $V = \{0, 1, 2\}$, and $E = \{(0, 1), (2, 1)\}$.

Figure 1: Left: Directed graph G, Right: Undirected graph G

Consider vertices $u = 0$ and $v = 2$. Clearly, there is no path from 0 to 2 or vice versa in the directed graph G, but there is a path $(0, 1, 2)$ from 0 to 2, and a path $(2, 1, 0)$ from 2 to 0, in the undirected graph.

Solution 2

From the definition of R, if uRv, then there is a path from u to v and vice versa, i.e., there is a path from v to u as well, which implies vRu. Hence, R *is symmetric*.

For going from u to u we need to do nothing, and so, the sequence which represents the path would contain just a single element u. Hence, R *is reflexive*.

Alternatively, let $(u_0 = u, u_1, ..., u_k = v)$ be a path from u to v, and let $(v_0 = v, v_1, ..., v_{k'} = u)$ be a path from v to u. Then, clearly, $u_0 = u, u_1, ..., u_k = v = v_0, v_1, ..., v_{k'} = u$ is a path (actually a cycle) from u to u. Note that here we have considered a cycle to be a path which is not consistent with the exact definitions.

Consider vertices u, v, w such that uRv and vRw, i.e., we have paths

$p_1 = (u_0 = u, u_1, ..., u_k = v)$

$p_2 = (v_0 = v, v_1, ..., v_{k'} = u)$

$p_3 = (v_0 = v, v'_1, ..., v'_m = w)$

$p_4 = (w_0 = w, w_1, ..., w_{m'} = v)$

Clearly, we have a path p_5 (using paths p_1 and p_3) from u to w and a path p_6 (using paths p_2 and p_4) from w to u given as-

$p_5 = (u_0 = u, u_1, ..., u_k = v = v_0, v'_1, ..., v'_m = w)$
$p_6 = (w_0 = w, w_1, ..., w_{m'} = v = v_0, v_1, ..., v_{k'} = u)$
Therefore, we have uRw. Hence, R *is transitive*.
Therefore, we can conclude that R is an equivalence relation.

Solution 3

Let us assume contrary that there exist vertices $a_1, b_1 \in V_1$ and $a_2, b_2 \in V_2$, such that we have edges $(a_1, a_2), (b_2, b_1) \in E$.
Since V_1 and V_2 are equivalence classes, a_1Rb_1 and a_2Rb_2, i.e., we have a path $p_1 = (b_1, u_1, u_2, ..., u_k, a_1)$ and a path $p_2 = (a_2, v_1, v_2, ..., v_{k'}, b_2)$.
Take any vertex $u \in V_1$, and any vertex $v \in V_2$. Since, V_1 and V_2 are equivalence classes, we have uRa_1 and vRa_2. So we have a path p_3 from u to a_1 and a path p_4 from a_2 to v. Hence we have a path from u to v via a_1 and a_2, as follows- go from u to a_1 through p_3, then from a_1 to a_2 along the edge (a_1, a_2), then from a_2 to v through p_4. Since u and v were arbitrarily chosen, we can say that there is a path from every vertex in V_1 to every vertex in V_2, we have the following lemma.
Lemma : Consider a directed graph which has strongly connected components A and B. If there exists an edge from a vertex in A to a vertex in B, then there exists a path from every vertex in A to every vertex in B.

Using above argument, we can similarly say, that since the edge (b_2, b_1) exists, there is a path from every vertex in V_2 to every vertex in V_1.
Therefore, in $V_1 \cup V_2$ there is a path from every vertex to every other vertex. Hence $V_1 \cup V_2$ is also an equivalence class (strongly connected component). Since we know that V_1 and V_2 are distinct, $V_1 \subset V_1 \cup V_2$, and $V_2 \subset V_1 \cup V_2$. Hence, they cannot be "maximal" subsets of V on which R is an equivalence relation, as required in the definition of equivalence classes. So, V_1 and V_2 cannot be equivalence classes, which is a contradiction. Hence our initial assumption must be false, i.e., it is not possible to have two edges one from a vertex in V_1 to a vertex in V_2, and other from a vertex in V_2 to a vertex in V_1 at the same time. Hence, all the edges from V_1 to V_2 must be in the same direction.

Solution 4

Consider a directed tree $T = (V, E)$. Now we make an important observation. If we count the number of incoming edges on each vertex, each edge will be counted exactly once, because every edge is incoming on exactly one vertex. Hence, we can say that -

$$\sum_{v \in V} indegree(v) = |E|$$

Now, for a directed tree, all vertices have indegree 1, except the root which has indegree 0. Therefore,

$$|E| = \sum_{v \in V} indegree(v) = 1 * (|V| - 1) + 0 = |V| - 1$$

Now, we know that if an undirected graph is connected and $|E| = |V| - 1$, then it is a tree. Therefore, a directed tree is also a tree if we ignore the edge directions. Hence, for any two *distinct* vertices $u, v \in V$, there must exist a unique path from u to v and vice-a-versa (considering edges as undirected). Let this path be $p = (u = u_0, u_1, u_2, ..., u_k = v)$, where for each $i = 0, 1, 2, ..., k-1$, either $(u_i, u_{i+1}) \in E$ or $(u_{i+1}, u_i) \in E$.
This is a path if we consider all edges to be undirected. This path may or may not be possible if the edges are directed. But atleast it is ensured, that no other path can exist in the directed tree, otherwise in the undirected version, the path will not be unique.
Now, note that this path from u to v is possible if and only if all the edges are in the same direction, i.e., either all the edges in path are of form (u_i, u_{i+1}) or all are of form (u_{i+1}, u_i), $i = 0, 1, 2, ..., k-1$. If we find an edge in path of the form (u_x, u_{x+1}) then, there cannot be a path from v to u, and if we find an edge in the path of the form (u_{y+1}, u_y) then, there cannot be a path from u to v.
Therefore, if a path from u to v exists along the directed edges, then all the edges in path are of form (u_i, u_{i+1}), hence there cannot be a path from v to u. Similarly, if a path exists from v to u along the directed edges, then all the edges in the path are of form (u_{i+1}, u_i), hence there cannot be a path from u to v.
Hence proved.

Solution 5

We are given that T_n = number of drawings of binary tree with n vertices. Consider a binary tree $G = (V, E)$, with n vertices. Let r be the root of G and u, v be the left child and right child of r respectively (if they exist). If we ignore the edges (r, u) and (r, v), then G can be seen as a combination of a root node, and two binary trees, one with u as root, and another with v as root. We call these two trees as the *left subtree* and *right subtree* of G respectively.

Using this idea, we can represent each drawing of G as a pair (D_L, D_R), where D_L is a drawing of the left subtree and D_R is a drawing of the right subtree. If in the original binary tree G, root has no left child, no such D_L exists. Similarly, if right child does not exist, no such D_R exists. To maintain consistency, we can allow an empty space to be a drawing of a binary tree with 0 vertices, we represent this drawing by D_0, and thus we have defined $T_0 = 1$.

Note that now, each such pair corresponds to exactly one drawing of G and each drawing of G corresponds to exactly one such pair, i.e., there is a bijection

between the set of all drawings of G and the set of all such pairs. Hence, total number of drawings of G = total number of pairs (D_L, D_R).
Now, consider all such pairs in which left subtree has j vertices, and right subtree has $n-j-1$ vertices.
Total number of such pairs, N_j (say) = number of drawings of left subtree $\times$ number of drawings of right subtree = $T_j T_{n-j-1}$
T_n = number of drawings of G = total number of pairs $(D_L, D_R) = \sum_{i=0}^{n-1} N_i = \sum_{i=0}^{n-1} T_i T_{n-i-1}$. Hence,

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

where, $n \geq 1$, and $T_0 = 1$.

Solution 6

Let us define a generating function for T_n as

$$G(x) = \sum_{i=0}^{\infty} T_i x^i$$

Note that for $n \geq 1$, coefficient of x^n in $xG^2(x)$ = coefficient of x^{n-1} in $G^2(x) = \sum_{i=0}^{n-1} T_i T_{n-i-1} = T_n$ = coefficient of x^n in $G(x)$. Hence, $xG^2(x)$ and $G(x)$ have same coefficients for x^n, for all $n \geq 1$, i.e., they differ only in the constant term. Let $xG^2(x) - G(x) = c$. Put $x = 0$, we get $c = -1$, and the equation becomes,

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

Note that this is a quadratic in $G(x)$. Using the quadratic formula, we get

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

or $2xG(x) = 1 \pm \sqrt{1-4x}$. We need to decide which sign to take. For this put $x = 0$. We know that $G(0) = T_0 = 1$. Therefore, $LHS = 0$, hence we need to take root with negative sign. So we have,

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

Let us expand $\sqrt{1-4x}$ using the binomial formula.

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

$\binom{1/2}{i} = \frac{1}{i!} \times \frac{1}{2} \times \frac{-1}{2} \times \frac{-3}{2} \times \frac{-5}{2} \times ... \times \frac{-(2i-3)}{2}$
$= \frac{1}{i!} \times \frac{1}{2^i} \times (-1)^{i-1} \times 1 \times 3 \times 5 \times ... \times (2i-3)$

$= \frac{(-1)^{i-1}}{2^i i!} \times \frac{1\times2\times3\times4\times5\times6\times\ldots\times(2i-4)\times(2i-3)\times(2i-2)\times(2i-1)\times(2i)}{2\times4\times6\times\ldots\times(2i)} \times \frac{1}{(2i-1)}$
$= \frac{(-1)^{i-1}}{2^i i!} \times \frac{(2i)!}{2^i i!} \times \frac{1}{(2i-1)}$
$= \frac{(-1)^{i-1}}{4^i(2i-1)}\binom{2i}{i}$
Therefore,
$\binom{1/2}{i}(-4x)^i = \frac{(-1)^{i-1}}{4^i(2i-1)}\binom{2i}{i} \times (-1)^i \times 4^i \times x^i = \frac{(-1)^{2i-1}}{(2i-1)}\binom{2i}{i} = \frac{-1}{(2i-1)}\binom{2i}{i}x^i$

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

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

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

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

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

Hence, we have

$$T_i = \frac{1}{i+1}\binom{2i}{i}$$

or

$$T_n = \frac{1}{n+1}\binom{2n}{n}$$

Solution 7

We are asked to prove that H is a tree. But this is not always true. Here we have a counter example.
Take a graph $G = (V, E)$, where $V = \{a, b, c, d, e, f, g, h, i, j, k, l\}$, and $E = \{(a, b), (b, c), (c, a), (d, e), (e, f), (f, g), (g, d), (h, i), (i, j), (j, k), (k, l), (l, h), (b, d), (c, h), (g, j)\}$

Note that $V_1 = \{a, b, c\}$, $V_2 = \{d, e, f, g\}$, $V_3 = \{h, i, j, k, l\}$ are the strongly connected components. Therefore, $V_H = \{1, 2, 3\}$.

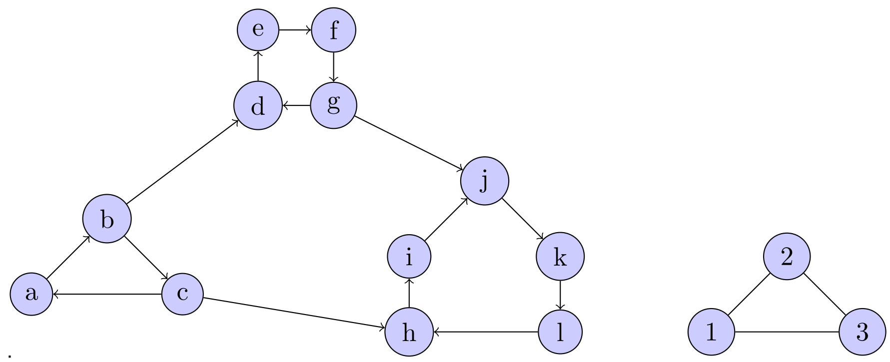

Figure 2: Left: Directed graph G, Right: Undirected graph H

In G, we have edge (b, d), $(b \in V_1, d \in V_2) \Rightarrow (1, 2) \in E_H$.
In G, we have edge (g, j), $(g \in V_2, j \in V_3) \Rightarrow (2, 3) \in E_H$.
In G, we have edge (c, h), $(c \in V_1, h \in V_3) \Rightarrow (1, 3) \in E_H$.
Therefore, $E_H = \{(1, 2), (2, 3), (1, 3)\}$
Now note that we have two distinct paths $1, 2$ and $1, 3, 2$ from 1 to 2, hence by definition, H is not a tree.

Solution 8

We are given an undirected graph, $G = (V, E)$. We need to prove the following-
(1) If it is a tree, then it is connected and does not have any cycle.
(2) If it is connected and does not have any cycle, then it is a tree.

Let us look at the definition of a tree first. "A connected undirected graph is a tree iff there exists a unique path from every vertex to every other other vertex in the graph."

Proof for (1): By definition of a tree, we know that the graph is connected. So we only need to prove that it does not have any cycle. Assume contrary, that we have a cycle $c = (u_0, u_1, u_2, ..., u_{k-1}, u_k = u_0)$ in the tree. Now consider the vertices u_0 and u_{k-1}. Clearly, we have a path from u_0 to u_{k-1}, given by $p = (u_0, u_1, u_2, ..., u_{k-1})$. But we also have a path along the edge $(u_{k-1}, u_k = u_0) = (u_0, u_{k-1})$ (since, the graph is undirected, edge direction does not matter, and this edge must exist because cycle c exists). Now, clearly we have two distinct paths from u_0 to u_{k-1}, hence the graph cannot be a tree, which is a contradiction. Hence a tree is connected and cannot have any cycle.

Proof for (2): Assume contrary that the graph G is connected and has no cycle, but it is not a tree. We have an undirected connected graph, if this graph has a unique path between every pair of vertices, then it would be a tree. But since we have assumed that it is not a tree, there must exist two vertices x

and y such that we have two distinct paths $p_1 = (x = x_0, x_1, x_2, ..., x_m = y)$ and $p_2 = (y = y_0, y_1, y_2, ..., y_n = x)$. Now if we move from x to y via p_1 and then from y to x via p_2, we get a cycle $c_0 = (x = x_0, x_1, x_2, ..., x_m = y = y_0, y_1, y_2, ..., y_n = x)$. This is a contradiction because we are given that there is no cycle in the graph. Hence, our assumption is wrong, and the graph is a tree.

Therefore, an undirected graph is a tree if and only if it is connected and does not have any cycle.

Solution 9

Consider a directed graph $G = (V, E)$.
Consider any cycle, $c = (u_0, u_1, u_2, ..., u_{k-1}, u_k = u_0)$. Let $U_i \subseteq V$ represent the strongly connected component containing the vertex u_i, $0 \leq i \leq k-1$.
For any $i = 0, 1, 2, ..., k-1$ we have the edge (u_i, u_{i+1}), hence by *Lemma* in *Solution 3*, (Kindly refer to Solution 3) we can say that there is a path from every vertex in U_i to every vertex in U_{i+1}.
Let $U_i = U_{i\ (mod\ k)}$, for $i \geq k$. Now take any arbitrary i, j. We make the following claim-

Claim: There exists a path from every vertex in U_i to every vertex in U_{i+j}.
Proof: We will prove this by induction on j. Base case $j = 1$, is proven (as mentioned above). Assume for j-1, i.e.,there exists a path from every vertex in U_i to every vertex in U_{i+j-1}. Also, we know that there exists a path from every vertex in U_{i+j-1} to every vertex in U_{i+j}. Take any vertex $x_0 \in U_{i+j-1}$. Clearly, we have a path from every vertex in U_i to every vertex in U_{i+j} as follows - take a path from vertex in U_i to x_0 and then from x_0 to vertex in U_{i+j}. Hence our claim is proven.

Now, since the choice of i and j is arbitrary, we have a path from every vertex in U_i to every vertex in U_{i+j}. Since we have defined U_i in a cyclic manner, $U_{i+j} \in S = \{U_0, U_1, U_2, ...U_{k-1}\}$ and, since all values of $j \geq 1$ are possible, U_{i+j} takes all values from S, some or other j. Hence we can say that for any i, j there exists a path from every vertex in U_i to every vertex in U_j.
Let

$$V' = \bigcup_{i=0}^{k-1} U_i$$

Note, that V' contains all the vertices of the cycle c. Take any two arbitrary vertices $x, y \in V'$. Let $x \in U_p$, $y \in U_q$. Then we know that there exists a path from x to y. Since, x and y are arbitrary, this is true for any pair of vertices x, y. Hence, V' is a strongly connected component. Note that for any given cycle c we have found a strongly connected component which contains all the vertices in the cycle. Therefore, all the vertices in a cycle of a directed graph lie in the same strongly connected component.

Solution 10

Statement 1: If an undirected graph has a spanning tree, then it is connected.
Proof: Consider any undirected graph $G = (V, E)$. Consider any spanning tree $T = (V, E')$ of G, Since, T is a tree, there exists a unique path from every vertex in V to every other vertex in V, and all the edges of this path belong to $E' \subseteq E$. Therefore, this path also exists in G. Hence, there exists a path between any two distinct vertices in G. Therefore, G is connected.

Statement 2: If an undirected graph is connected, then it has a spanning tree.
Proof: If there is a unique path between any two distinct vertices in G, then G is itself a spanning tree, and we are done. Otherwise, take any two vertices u and v such that there exist atleast two distinct (non-overlapping) paths between u and v. By non-overlapping paths we mean that they must not have any common edge. Let these paths be $p_1 = (u_0 = u, u_1, u_2, ..., u_k = v)$, and $p_2 = (v_0 = u, v_1, v_2, ..., v_m = v)$. Remove the edge (u, u_1) from G and let the resulting subgraph be G_1. Now we will prove that G_1 is also connected. Take any two vertices a, b in G_1, look at the path p between them in G. If it does not contain the edge (u, u_1), take this path. If it contains the edge (u, u_1), take the following alternative path - move from a to u along the edges of path p, then move from u to v along the path p_2, then move from v to u_1 along the edges of the path p_1 then continue along the path p to go from u_1 to b.
Therefore, G_1 is also connected. Now, if there is a unique path between any two distinct vertices in G_1, then G_1 is a spanning tree of G, and we are done. Otherwise repeat the same process with G_1 to get G_2. If G_2 is a spanning tree, we are done, otherwise keep repeating the process to get $G_3, G_4, ...$ and so on. Since the number of edges are finite, and each time we are removing atleast one edge (keeping the vertices untouched), it is ensured that at some point we will have to stop at some G_n. This G_n will be our spanning tree.

Hence we have proven that an undirected graph has a spanning tree iff it is connected.

Solution 11

Since we are hereby talking about water networks, it is obvious that we need our subgraphs to be connected. Let S_G represent the set of all connected subgraphs of G. Let T_G be the set of all spanning trees of G. Let M_G be the set of all minimum spanning trees of G. Obviously, $M_G \subseteq T_G \subseteq S_G$. Consider a minimum weight (connected) subgraph W of G. Now, we know that $W \in S_G$. Suppose $W \in S_G \backslash T_G$, i.e., W is not a spanning tree. Since W is connected, (using the statement of question 10) it must have a spanning tree. Let it be H. Let $W = (V_W, E_W)$ and $H = (V_H, E_H)$. Since H is a spanning tree of W, $V_H = V_W$ and $E_H \subseteq E_W$. And since W is not a spanning tree, $E_H \subset E_W$. Since weight of a subgraph is the sum of weights of all its edges, and edges

of H are a subset of edges of W. This implies that weight of $H <$ weight of W, which is not possible because W is a minimum weight subgraph. Hence, $W \notin S_G \backslash T_G \Rightarrow W \in T_G$, i.e., W must be a spanning tree.
Now, suppose $W \in T_G \backslash M_G$, i.e., W is not a minimum spanning tree. Now take any $A \in M_G$. By definition of a minimum spanning tree, weight of $A <$ weight of W (because W is a spanning tree but not a minimum spanning tree). This again is not possible because W is a minimum weight subgraph. Hence, $W \notin T_G \backslash M_G \Rightarrow W \in M_G$, i.e., W must be a minimum spanning tree.
Therefore, we have proven that a minimum weight subgraph must be a minimum spanning tree.

Solution 12

Let the given weighted graph be $G = (V, E, w)$. Let us call the graph formed after i^{th} iteration of the algorithm be $T_i = (U_i, E'_i, w)$, where U_i is the set of vertices U (mentioned in the algorithm) obtained after adding the vertex in step 4, and E'_i is the set of edges after are adding the edge in step 4. We have $T_0 = (U_0 = \{v\}, E'_0 = \phi, w)$.
For instance, if v_i, e_i are the vertex and edge added in the i^{th} iteration, then $U_i = \{v, v_1, v_2, ..., v_i\}, E'_i = \{e_1, e_2, ..., e_i\}$.
Also, let $G_i = (U_i, E_i, w)$, be the graph formed by all the vertices in U_i and has all the possible edges from G, i.e., $E_i = E \cap U_i \times U_i$. Clearly, T_i is a subgraph of G_i.
Since in each iteration, we are adding atleast one vertex in U from V, and $|V| = n$ (*say*) is finite, the loop will surely end. It ends after $n - 1$ (*say*) iterations. Note that $G_{n-1} = G$. So, we need to prove that T_{n-1} is a minimum spanning tree of $G = G_{n-1}$.
First we will show that T_i *must be a spanning tree of* G_i $(i \leq n - 1)$.

Note that T_0 is connected (it is a single vertex with no edge), and if T_{i-1} is connected, then after adding the next vertex (*say*) x and an edge from a vertex *say* y in T_i to x, T_i will also be connected (for going from any vertex to x, first go to y, then go from y to x, and for all other pair of vertices, we already have a path because T_{i-1} is connected). Hence by induction of i, T_i *must be connected* for all $i \leq n - 1$.
Now, note that for T_0, $|E'_0| = |U_0| - 1$, and if for T_{i-1}, $|E'_{i-1}| = |U_{i-1}| - 1$, then after adding the next vertex and the corresponding edge, we will have for T_i, $|E'_i| = |E'_{i-1}| + 1 = |U_{i-1}| - 1 + 1 = |U_i| - 1$ (because, $U_i = U_{i-1} \cup \{v\}, E'_i = E'_{i-1} \cup \{e\}$, where v and e are the vertex and edge added in i^{th} iteration). Hence, again by induction on i, T_i *has* $|E'_i| = |U_i| - 1,\ (i \leq n-1)$. Since T_i is connected too, it must be a spanning tree for all $i \leq n - 1$.
Now to prove that T_{n-1} is a minimum spanning tree, we make the following assertion-
Assertion $P(i)$: T_i is contained by a spanning tree of G, i.e., all the vertices and edges of T_i are in that spanning tree.

If we prove that $P(n-1)$ is true, then we have that T_{n-1} is contained by a spanning tree (*say*) $T = (V, E_T, w)$ of G. But since $|U_{n-1}| = |V| = |E_T| + 1 = |E'_{n-1}| + 1$, we must have $T_{n-1} = T$, i.e., T_{n-1} is a spanning tree of G. So, we only need to prove that $P(n-1)$ is true. We will do this by induction on i.
Base Case: $P(0)$ is true. If $T = (V, E_T, w)$ is a minimum spanning tree of G, then $U_0 = v \subseteq V$, and $E'_0 = \phi \subseteq E_T$. Therefore, T_0 is contained by T.
Assume $P(i-1)$ to be true. We need to prove that $P(i)$ is true. Just before the i^{th} iteration, we have the graph $T_{i-1} = (U_{i-1}, E'_{i-1}, w)$. Let the vertex added in i^{th} iteration be v_i, and the edge added be $e_i = (u, v_i)$, where, $v_i \notin U_{i-1}$ and $u \in U_{i-1}$. So, we have $U_i = U_{i-1} \cup \{v_i\}$ and $E'_i = E'_{i-1} \cup \{e_i\}$. Let $T = (V, E_T, w)$ be a minimum spanning tree of G. Now we have two cases -
CASE 1: $e_i = (u, v_i) \in E_T$.
In this case, since we know that $E'_{i-1} \subseteq E_T$ (by induction hypothesis), we can say that $E'_i = E'_{i-1} \cup \{e_i\} \in E_T$. And obviously, $U_i \subseteq V$. Therefore in this case we can say that T_i is contained by a minimum spanning tree of G (T in this case).
CASE 2: $e_i = (u, v_i) \notin E_T$.
In this case, since, T is a minimum spanning tree, it is connected. So there must exist an edge $e = (u', v_i) \in E_T$, such that $u' \in U_{i-1}$. Remove this edge e from T and add the edge e_i. Let us call this new graph $T' = (V, E'_T, w)$. Note that same as in CASE 1, T_i is contained by T'. Note that, T' is connected, because for all the paths containing the edge e, we can now find an alternative path containing the edge e_i (if we need to go from u' to v_i, we can first go from u' to u then from u to v_i). Also since the number of edges is unchanged, $|E'_T| = |E| = |V| - 1$. Hence, T' is also a spanning tree. Now, in our algorithm, we compare all the edges from any vertex in U_{i-1} to v_i and choose the edge with minimum weight. Since we chose e_i and rejected e, $weight\ of\ e_i \leq weight\ of\ e \Rightarrow weight\ of\ T' \leq weight\ of\ T$ (since all other edges in T and T' are same). Now, since T was a minimum spanning tree, T' must also be a minimum spanning tree. Hence, in this case too, we can say that T_i is contained by a minimum spanning tree of G (T' in this case).
Hence, $P(i)$ is also true.
Therefore, we have proven by induction that $P(i)$ is true for all $i \leq n-1$. Hence, $P(n-1)$ is true $\Rightarrow T_{n-1}$ is a minimum spanning tree of G (as argued above). This completes our proof.

Solution 13

Let $G = (V, E)$ be the given undirected weighted connected graph. Consider any arbitrary tour T. We will represent the length of any tour T as l_T. In tour T we traverse some edges. Each edge may be traversed one or more than one time. Let $E_T \subseteq E$ be the set of all the edges traversed atleast once in the tour T. Length of the tour, l_T is given by sum of all the edges, where if an edge is traversed multiple times, it is counted that many number of times. Consider the subgraph $G_T = (V, E_T)$. Since in a tour T, we have visited every vertex atleast once, it is ensured that G_T is connected (for any two vertices, we can find a path between them along the edges in E_T). We know that, minimum spanning tree $T_0(say)$ is the minimum weight (connected) subgraph. Therefore, weight of $T_0 \leq$ weight of G_T. Also, since each edge in G_T appears once or more than once in tour T, we can say that weight of $G_T \leq l_T$. Hence we have, weight of $T_0 \leq$ weight of $G_T \leq l_T \Rightarrow$ weight of $T_0 \leq l_T$. Now, since T was arbitrarily chosen, this is true for all tours. So, it must also be true for the minimum length tour. In other words, weight of minimum spanning tree of a graph is at most the minimum length tour.

Solution 14

We need to prove that minimum length tour $\leq$ weight of minimum spanning tree. For this let us take a graph $G = (V, E, w)$. Let P_0 be the minimum length tour, and let T be a minimum spanning tree of G. We will represent the length of any tour P by l_P.
First we will prove a few properties needed later in our proof.

Property 1: Every tree with more than 1 vertices has a vertex with $degree = 1$.
Proof: If any vertex in a tree (with more than 1 vertices) has degree 0, then that vertex has no edge, and thus no path can exist between that vertex and any other vertex in the tree, i.e., it will not be connected. Therefore, degree of all vertices ≥ 1. Suppose there exists a tree $T = (V, E)$, for which degree of all vertices ≥ 2. We know, that sum of degrees of all the vertices $= 2|E|$ and $|E| = |V| - 1$. Then, $\sum_{v \in V} degree\ of\ v \geq 2|V| \Rightarrow 2|E| \geq 2|V| \Rightarrow |E| \geq |V| \Rightarrow |V| - 1 \geq |V| \Rightarrow -1 \geq 0$, which is a contradiction. Hence, no such tree exists. Therefore, there exists a vertex in every tree (with more than 1 vertices) which has degree 1.

Property 2: If we remove a vertex with degree 1 from a tree (such a vertex exists from Property 1), the resulting graph is also a tree.
Proof: Let $T = (V, E)$ and $T' = (V', E')$ be the initial tree and the reduced graph respectively. Firstly, we need to prove that T' is connected. Let the removed vertex be v, and the removed edge be $e = (u, v)$. Now, any path, starting from any vertex other than v, in T, containing the edge e must end in v. Because v has only 1 edge and if we use that edge in our path to reach v, then we cannot go any further, since v has no other edge, and same edge cannot be traversed again in a path (we cannot use e to go out of v otherwise, how did we reach v in the first place, if we did not use v). Therefore, it is ensured, that if we take two vertices in T' and look at the path between them in T, e will not lie on that path, otherwise that path would v at one end which is not possible, since we took the end points on T' (which does not contain v). Now, all other edges in T are untouched, hence in T' we have a path from every vertex to every other vertex. So T' is connected.
Now, since T is a tree, $|E'| = |E| - 1 = (|V| - 1) - 1 = |V'| - 1 \Rightarrow T'$ is a tree.

Property 3: Remove any vertex with degree 1 from T and G (such a vertex exists by Property 1). Let the resulting graphs be T' and G' respectively. Then, T' will be a minimum spanning tree of G'.
Proof: Firstly, note that T' is a subgraph of G'. Also by Property 2, T' is a tree. Therefore, T' is a spanning tree of G'. Let the removed vertex be v, and the removed edge be $e = (u, v)$. Consider any arbitrary spanning tree T_1' of G'. Add the vertex v and edge e in T_1' to get T_1. Note that T_1 would be a spanning tree of G. Now, $weight\ of\ T_1 \geq weight\ of\ T \Rightarrow weight\ of\ T_1' + weight\ of\ e \geq weight\ of\ T' + weight\ of\ e \Rightarrow weight\ of\ T_1' \geq weight\ of\ T'$. Since T_1' was arbitrarily chosen, it is true for all spanning trees of G'. This implies that T' is a minimum spanning tree of G'.

Property 4: Given a connected graph $G = (V, E)$. There exists a tour P on G such that $length\ of\ P = 2 \times weight\ of\ T$, where T is a minimum spanning tree of G.
Proof: We will prove this by induction on $|V|$.
BASE CASE: $|V| = 2$ — We have two vertices $u, v \in V$, and exactly one edge $e = (u, v)$. Let us represent a tour as a sequence of vertices. Note that in this case, G itself is a minimum spanning tree. We can chose $P = (u, v, u)$. $length\ of\ P = 2 \times weight\ of\ e = 2 \times weight\ of\ G = 2 \times weight\ of\ T$ (in this case T and G are same).
Assume for $|V| = n$. Take a graph $G = (V, E, w)$, with $|V| = n + 1$, and let $T = (V, E_T, w)$ be a minimum spanning tree of G. By, Property 1, we have atleast one vertex in G, with degree 1. Remove that vertex from T and G. Let the removed vertex be v, and the removed edge be $e = (u, v)$. Let the resulting graphs be $T' = (V', E_T', w)$ and $G' = (V', E', w)$ respectively. Then by Property 3, T' is a minimum spanning tree of G'. Now note that $|V'| = n$. Therefore, by induction hypothesis, we have a tour P', such that $length\ of\ P' = 2 \times weight\ of\ T'$. Now, we can create a tour P on G using P', as follows— Since P' is a tour on G', and $u \in V'$, u must appear atleast once in the sequence of vertices representation of P'. Let the first occurrence of u be after $k - 1$ terms. So, $P' = (v_1, v_2, ..., v_k = u, v_{k+1}, ..., v_m)$. Now, let $P = (v_1, v_2, ...v_k = u, v, u, v_{k+1}, ..., v_m)$. Note that P is a tour on G. Also, $length\ of\ P = length\ of\ P' + 2 \times weight\ of\ e = 2 \times weight\ of\ T' + 2 \times weight\ of\ e = 2 \times (weight\ of\ T' + weight\ of\ e) = 2 \times (weight\ of\ T)$, i.e., we have a tour P on G, such that $length\ of\ P = 2 \times (weight\ of\ T)$.
This completes the proof by induction.

Now, if P_0 is a minimum length tour and P_1 is any tour, then by definition, $length\ of\ P_0 \leq length\ of\ P_1$. By Property 4, we know that there exists a tour P, such that $length\ of\ P = 2 \times (weight\ of\ minimum\ spanning\ tree)$. Therefore we have, $length\ of\ P_0 \leq length\ of\ P = 2 \times (weight\ of\ minimum\ spanning\ tree) \Rightarrow length\ of\ P_0 \leq 2 \times (weight\ of\ minimum\ spanning\ tree)$. Hence proved, that minimum length tour can be atmost the weight of minimum spanning tree.