Student Information

Full Name : Ceydanur Köylü
Id Number : 2633386

Answer 1

a) Yes, there exists an Eulerian Circuit in the given graph. We can show this by using the theorem **"A connected multigraph with at least two vertices has an Euler circuit if and only if each of its vertices has even degree."** (Theorem 1, given in page 696 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

Each vertex in the graph indeed has an even degree such that:

	Degree of Each Vertex
a	2
b	2
c	4
d	4
e	4
f	2
g	4
h	4
i	4
j	4
k	2
l	4
m	2

An example of such an Eulerian Circuit is:

$$b, c, d, e, j, i, l, h, i, d, h, g, c, a, e, f, j, m, l, k, g, b$$

b) No, there is no such Eulerian Path (which is not a circuit) in the given graph. We can show this by using the theorem **"A connected multigraph has an Euler path but not an Euler circuit if and only if it has exactly two vertices of odd degree."** (Theorem 12, given in page 697 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

I have already shown in part (a), that all vertices in the graph have even degrees, so by the theorem above, there exists **no such Eulerian Path.**

Also we can reach this conclusion by ourselves, since to form a path which is not a circuit, we need end and start vertices. For the end vertex, the number of times we enter is one more than we exit; and for the start vertex the number of times we exit is one more than we enter. Which means the degree of such vertices would be $n + n + 1 = 2n + 1$ (where n is the number of times we exit/enter) which is odd. So we need two vertices with odd degrees to find such a path, which is not the case for this graph.

c) A Hamiltonian Circuit is a simple circuit where we go through every vertex in the graph only once, and come back to the beginning vertex (Definition 2, given on page 698 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition). **There is no such Hamiltonian Circuit for this graph.** Let's assume such a circuit exists and try to get a contradiction.

Note that since every vertex must be visited and we must construct a circuit with the path; we must use an edge to enter a vertex, and one to exit it (this is true for all vertices, even the beginning/ending vertex must obey this rule). Then after this process, all the other edges on that vertex would become unusable, since by using them we would come to the vertex again which is forbidden.

We must notice that for the vertices with degree 2, the only way to add them to the Hamiltonian Circuit is to include both of the edges that connect to them, since otherwise we wouldn't be able to visit and exit these vertices.

I have listed the degrees of all vertices in part (a), and by using this listing we can paint the edges of all vertices which have a degree of 2. So we can say that the edges painted in green must be included in the Hamiltonian Circuit, such that (note that since we have to include these green edges in the Hamiltonian Circuit, we necessarily visit the vertices $b, c, a, e, f, j, m, l, k, g$. In the below figure, I have denoted these forcibly visited vertices with the color green) :

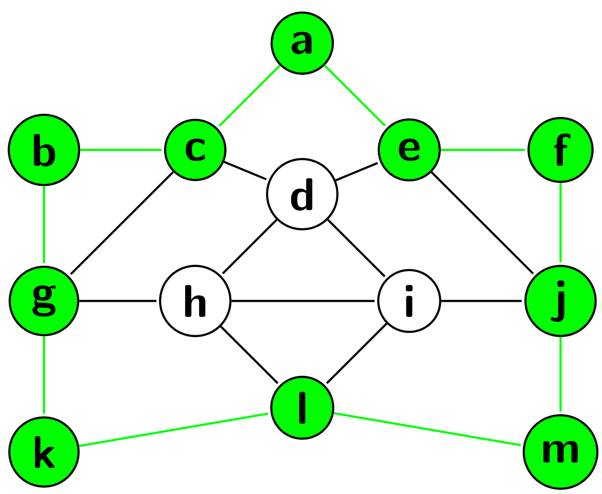


Figure 1: Edges that must be included

Note that since we have to include these green edges in the Hamiltonian Circuit, we necessarily visit the vertices $b, c, a, e, f, j, m, l, k, g$. In the below figure, I have denoted these forcibly visited vertices with the color green.

By following this process as I explained above, if we include these green edges in the Hamiltonian Circuit, we need to get rid of all the other edges that the visited green vertices have; as they will become unusable (since we cannot visit the already visited vertices). In the below graph I have shown the unusable edges by the color red:

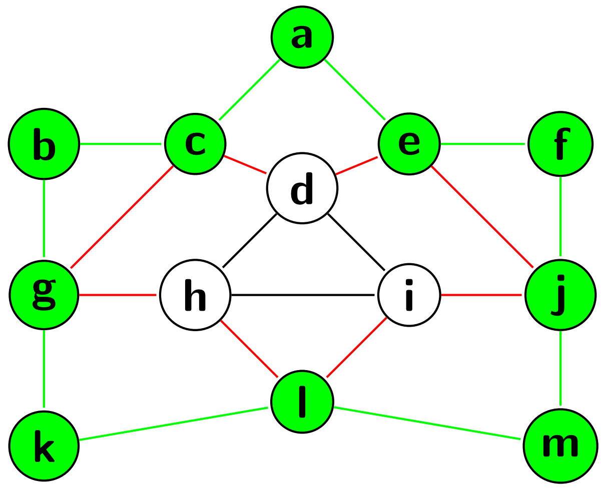


Figure 2: Edges which we cannot use in the circuit

Finally, it is clear to see that there is no way to connect the inner circuit d, h, i to the outer circuit we have constructed, as we are forbidden to use the red edges and we must include the green edges in the path. So, there is no possible Hamiltonian Circuit for this graph.

d) A Hamiltonian Path is a path where we go through every vertex in the graph only once (Definition 2, given in page 698 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

So yes, there exists a Hamiltonian Path (which is not a circuit) for this graph. We can show an example of such a path as:

$$b, c, a, e, f, j, i, d, h, g, k, l, m$$

Note that in this path we visit every vertex only once, and the beginning and end vertices are different, hence not a circuit. (Which is impossible to have anyway, as proved in part (c))

e) The given graph G's chromatic number $\chi(G)$ is equal to 3, as the minimum number of colors we can use to color G, such that no adjacent vertices share the same color, is 3.

Note that we cannot color this graph by using 2 or less colors, since there are cliques of 3 vertices in G (such as e, f, j or b, c, g) for which we cannot color by using two colors. As all of the vertices in these K_3 subgraphs are adjacent to each other, we need 3 colors to color these subgraphs.

The graph coloring using 3 colors is:

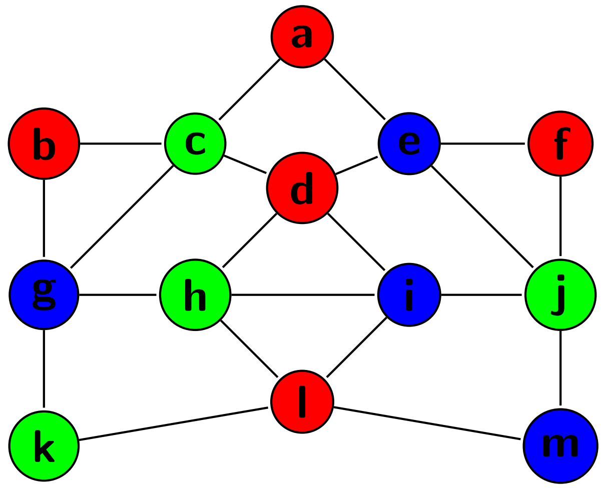


Figure 3: Graph coloring with 3 colors

f) A simple graph G is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color, in other words, G is bipartite if and only if it is 2−colorable (Theorem 4, given on page 657 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition)

In part (e), I have shown that G is not 2−colorable since it contains subgraphs which are isomorphic to K_3. So G is not bipartite.

To make G bipartite, we need to get rid of the subgraphs of G which construct K_3 isomorphic structures. To do this, we need to remove at least 3 edges, since there are 4 such cliques present, which are destructible by removing some essential edges, which are $\{b, g\}, \{f, j\}, \{h, i\}$.

Then the graph becomes 2−colorable, thus bipartite:

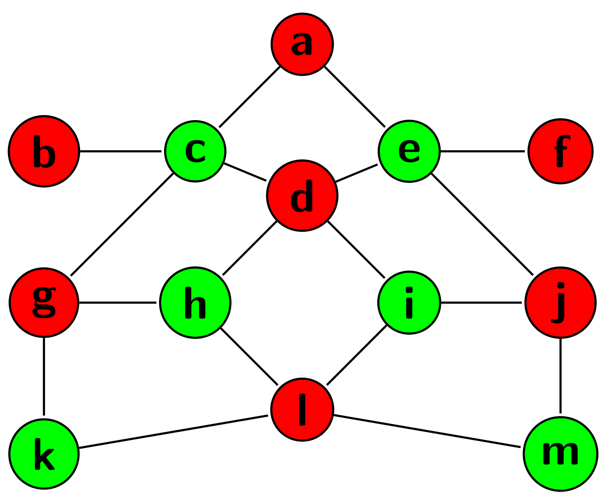


Figure 4: Graph coloring with 2 colors

g) Note that for a subgraph G' of G to be a complete graph, it needs 4 adjacent vertices, each having a degree of at least 3. I have already listed the degrees of each vertex in part (a), so by using this information we can exclude all vertices which have a degree less than 3, and start constructing our subgraph:

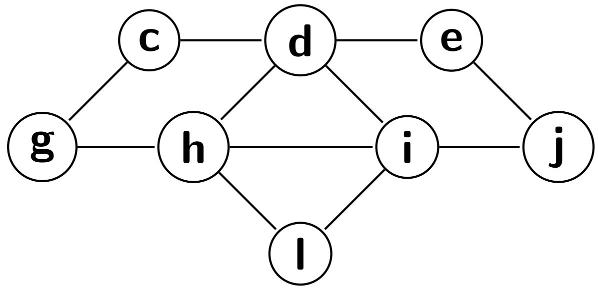


Figure 5: A subgraph of G, which only includes vertices with degrees ≥ 3

However, it is clear to see that there is no complete subgraph which 4 nodes of the above G' as there are no 4 nodes where each share an edge between the distinct chosen pairs.

So to construct a K_4 out of the above given nodes, we need to add at least one edge, which is $\{d, l\}$. Note that if we add this edge, our initial graph G becomes modified such that:

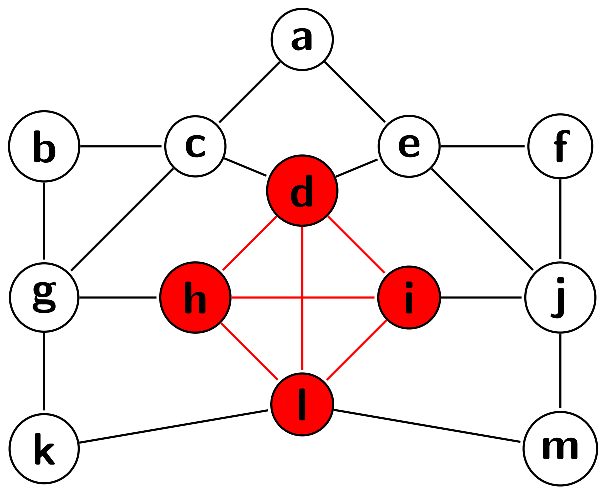


Figure 6: New G with a subgraph of K_4

Answer 2

Yes, the graphs G and H are isomorphic. To start proving this, we can first check the graph invariants to see if there is any obvious error. If we say $G = (V_1, E_1)$ and $H = (V_2, E_2)$, we can easily see that:

$$|V_1| = |V_2| = 8, \quad |E_1| = |E_2| = 16, \quad deg(u) = deg(v) = 4, \quad u \in V_1, v \in V_2$$

Then since there is no obvious error to prove they are isomorphic, we can go further and try to find a function f, which is one-to-one and onto, defined from the vertex set of G to the vertex set of H; with the property if a and b are vertices adjacent to each other in G, $f(a)$ and $f(b)$ are vertices adjacent to each other in H (Definition 1, given on page 672 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition). We can define this function as:

$x \in V_1$	$f(x) \in V_2$
a	a'
b	g'
c	e'
d	c'
e	h'
f	b'
g	f'
h	d'

Then, if we can show that the adjacency matrix of G can be represented with $f(x) \in V_2$ where $x \in V_1$, it means that G and H are isomorphic (as mentioned above, isomorphic graphs are defined with the property if a and b are vertices adjacent to each other in G, $f(a)$ and $f(b)$ are vertices adjacent to each other in H). Adjacency matrix of G can be represented as:

$$M_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} \begin{array}{c} \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

By using the function described as above we can write:

$$M_G = \begin{array}{c} \\ a' \\ g' \\ e' \\ c' \\ h' \\ b' \\ f' \\ d' \end{array} \begin{array}{c} \begin{array}{cccccccc} a' & g' & e' & c' & h' & b' & f' & d' \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Notice that the above matrix is just a reorganization of:

$$M_H = \begin{array}{c} \\ a' \\ b' \\ c' \\ d' \\ e' \\ f' \\ g' \\ h' \end{array} \begin{array}{c} \begin{array}{cccccccc} a' & b' & c' & d' & e' & f' & g' & h' \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Hence we found a function f, which is one-to-one and onto, defined from the vertex set of G to the vertex set of H; with the property if a and b are vertices adjacent to each other in G, $f(a)$ and $f(b)$ are vertices adjacent to each other in H, thus we have proved that H and G are isomorphic.

Answer 3

Before we begin, it is useful to recall that: A simple graph G is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color, in other words, G is bipartite if it is $2-$colorable (Theorem 4, given on page 657 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition)

a) For $n \geq 3$ C_n cycle graphs, my claim is that $\chi(C_n) = 3$ if n is odd, and $\chi(C_n) = 2$ if n is even. To prove this claim let's first look at the case where n is even.

For $n \geq 4$ C_n, where n is even, let's write the vertices in the cycle as a chain such that:

$$x_1, x_2, x_3 \cdots x_{n-1}, x_n \text{ where } x_1 \text{ and } x_n \text{ are adjacent to each other}$$

Starting from x_1, let's assign the color Black to even indexed vertices, and the color Red to odd indexed vertices. Since every vertex only has two adjacent vertices, this process ensures that the vertices will have different colors compared to their neighbors. By applying this rule we assign x_1 Red and x_n Black, which means we don't have to assign a third color at all. Thus, $\chi(C_n) = 2$ if n is even.

Also by the above explanation, since $\chi(C_n) = 2$, the graph C_n $n \geq 4$ is bipartite if n is even.

For $n \geq 3$ C_n, where n is odd, let's write the vertices in the cycle as a chain such that:

$$x_1, x_2, x_3 \cdots x_{n-1}, x_n \text{ where } x_1 \text{ and } x_n \text{ are adjacent to each other}$$

Starting from x_1, let's assign the color Black to even indexed vertices, and the color Red to odd indexed vertices. Similar to the above case, since every vertex only has two adjacent vertices, this process ensures that the inner vertices will have different colors compared to their neighbors. However, by applying this rule we assign x_1 and x_n both Black, which means we have two adjacent nodes that have the same color. To solve this problem let's assign a third color, say Yellow to x_n. Thus, we have a graph where each vertex has a color different than its neighbors, hence $\chi(C_n) = 3$ if n is odd.

Also by the above explanation, since $\chi(C_n) = 3 \neq 2$, the graph C_n where $n \geq 3$ is not bipartite if n is odd.

b) For $n = 1$, we have a graph with 1 vertex, Q_1, which has a chromatic number $\chi(Q_1) = 1$ as seen clearly. Also Q_1 is not bipartite since $\chi(Q_1) \neq 2$.

For $n > 1$, Q_n is the graph where each node is represented by a distinct binary string of length n, and two nodes are adjacent if and only if their binary string sequences differ by exactly 1 digit. My claim is that $\chi(Q_n) = 2$ for $n > 1$.

To prove this claim, I will use the above theorem which states a graph G is bipartite if and only if it is $2-$colorable. First of all, for the graph $Q_n = (V, E), n > 1$ let's define two sets of vertices V_o and V_e such that; V_o is the set of vertices for which the associated binary string has an odd number of 0's in it, and V_e is the set of vertices for which the associated binary string has an even number of 0's in it. Note that $V_e \cup V_o = V$, also V_e and V_o are mutually exclusive by their definition.

Notice that for every pair of distinct elements in V_e and V_o, their associated binary strings differ by at least 2 elements, since $\{1, 3, 5, 7, \cdots\}$ are the possible numbers of 0's in each corresponding element's binary string in V_o, and $\{0, 2, 4, 6, \cdots\}$ are the possible numbers of 0's in each corresponding element's binary string in V_e. Hence, it is impossible for a distinct pair of elements both chosen from either V_o or V_e, to have associated binary strings differing by exactly 1 digit. This means that none of the vertices in the set of V_e is adjacent to each other, similarly none of the vertices in the set of V_o is adjacent to each other, (However, it is of course possible to choose a pair of vertices $\{v_1, v_2\}, \quad v_1 \in V_o, v_2 \in V_e$ which are adjacent to each other.)

So, since we can partition all of the vertices in two sets such that there are no nodes that are adjacent to each other inside these sets; we can say that Q_n is a bipartite graph. Also by the above theorem, we are able to say that $\chi(Q_n) = 2$. ($\chi(Q_n) \neq 1$ since $n > 1$.)

Answer 4

a) I will use Kruskal's Algorithm to find the minimum spanning tree. To use the algorithm, I will first sort all edges from minimum to maximum weight such that:

$$[\{a, b\}, \{e, c\}, \{c, f\}, \{e, f\}, \{a, d\}, \{c, b\}, \{e, b\}, \{d, e\}, \{a, e\}]$$

In Kruskal's Algorithm, first of all we take all vertices as connected components consisting of one node, and we add edges if they do not form any cycles (from minimum to maximum weight) to our subgraph. We stop this process when we add $n - 1$ total edges to our subgraph(Algorithm 2, given on page 800 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

So to start with, let's denote our MST as G_{MST}, and add the edge $\{a, b\}$, thus get a connected component consisting of vertices a and b. We can continue this process and add $\{e, c\}, \{c, f\}$ to our spanning tree, forming another connected component consisting of c, e, f. However we cannot continue by adding $\{e, f\}$ to our MST (note that this is caused by our initial ordering of edges, since $\{e, c\}, \{c, f\}, \{e, f\}$ are all of same weight, we could have included $\{e, f\}$ and excluded $\{c, f\}$, which would still be correct for constructing a MST), since adding this edge would cause a cycle, and we do not want any new edges forming between the vertices of already connected components. With this process, we would add the edges to G_{MST} in this order:

$$\{a, b\}, \{e, c\}, \{c, f\}, \{a, d\}, \{c, b\}$$

b) By using the G_{MST} we constructed above, we can represent our subgraph as:

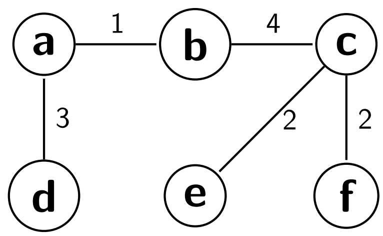


Figure 7: G_{MST} constructed with Kruskal's Algorithm

c) As I have mentioned above, when we construct our G_{MST}, we don't have a special ordering for edges with the same weights which can be added to our subgraph such that they don't form a cycle. An example of this situation can be observed for the edges $\{e, c\}, \{c, f\}, \{e, f\}$, as they all have the same weight factor and we need to choose 2 of them to construct our MST. This choice is arbitrary, as it doesn't affect the cost of our tree and all edges are valid edges which can be added to our MST.

So, our MST is not unique, we could have constructed a different and yet valid MST by including $\{c, f\}, \{e, f\}$ instead of $\{e, c\}, \{c, f\}$. The new G_{MST} then would be:

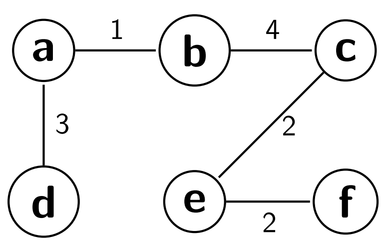


Figure 8: Another G_{MST} constructed with Kruskal's Algorithm

Answer 5

a) Firstly I will borrow some definitions from the textbook to be clear about what I'm referring to in the below sections: "The **level** of a vertex v in a rooted tree is the length of the unique path from the root to this vertex. The level of the root is defined to be zero. The **height** of a rooted tree is the maximum of the levels of vertices." (Balanced m-ary Trees, given on page 753 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

For a full binary tree of height k, it is clear to see that for each level l, where $0 \leq l \leq k$, the number of nodes at each level is equal to 2^l. To prove this claim, since each branch node has exactly 2 children, we can show this with induction. Base case is $l = 0$, there is only 1 node at the level, $2^0 = 1$ Then the Inductive Hypothesis assumes this claim holds for $l = k, l \geq 0$, such that the number of nodes at level k is 2^k. For the $l = k+1$ case, we can say that in this level, there exists only the children of the previous level's branch nodes. Then by using the Inductive Hypothesis, the number of nodes in $l = k+1$ is $2 * 2^k = 2^{k+1}$.

Then the total number of nodes in such a binary tree with height k is:

$2^0 + 2^1 + 2^2 + \cdots + 2^{k-1} + 2^k = \displaystyle\sum_{i=0}^{k} 2^i$ we can represent this number with a binary expansion:

$$\sum_{i=0}^{k} 2^i = (11111 \cdots 11)_2 \text{ with k 1's. If we add 1 to this number:}$$

$$\sum_{i=0}^{k} 2^i + 1 = (10000 \cdots 00)_2 \text{ with k 0's. Note that this number corresponds to:}$$

$$\sum_{i=0}^{k} 2^i + 1 = 2^{k+1}, \quad \sum_{i=0}^{k} 2^i = 2^0 + 2^1 + 2^2 + \cdots + 2^{k-1} + 2^k = 2^{k+1} - 1$$

Then we can set $n = 2^{k+1} - 1$. Also note that for a full binary tree, the number of leaves is exactly the number of nodes at the maximum level, which is 2^k in our case. So we can write the number of leaves as:

$$2^k = \frac{n+1}{2} = \text{ the number of leaves in a full binary tree.}$$

b) A tree is a connected graph which contains no cycles, by definition. We also know that in a tree, there exists a unique simple path between every pair of distinct elements (Theorem 1, given on page 746 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

Then, I claim that all trees are bipartite graphs, and hence, their chromatic number $\chi(T) = 2$ for $n > 1$ where n is the number of vertices in the tree (Theorem 4, given on page 657 of Kenneth Rosen's "Discrete Mathematics and Its Applications", 7^{th} Edition).

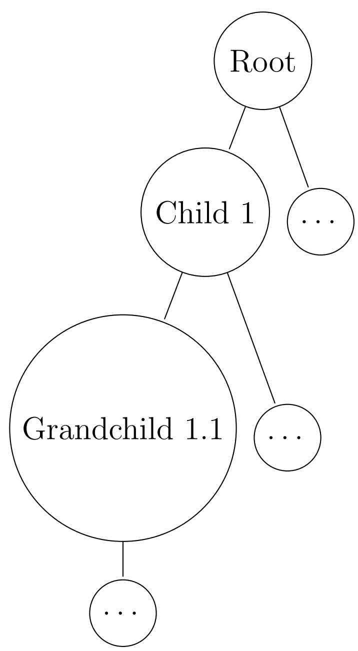
To prove all trees that have a number of vertices $n > 1$ are bipartite, let's first pick an arbitrary node from the tree and call it v_0. By the above explanation, we know that there must exist a simple and unique path between every other element of this graph and v_0. Then, we can try and visit every other element in this tree from this node.

As we go and visit every other node, let's put them in the sets A and B such that if the path length to that particular node is odd numbered we put it in the set A, and if it is even numbered we put it in set B. Also, we will put v_0 in set B since we can take its path length to be 0.

Note that by the above given theorems, two adjacent vertices in the tree will certainly be in different sets, since their path length will differ by 1 (since the paths are unique). So no neighboring elements will be in the same set. Hence, I showed that we can take the vertices of a graph and separate it into two sets such that no two adjacent nodes are in the same set, which means trees are indeed bipartite.

Since a graph G is bipartite if and only if it is $2-$colorable, we can say that trees' chromatic number is $\chi(T) = 2$ for $n > 1$. For the trivial case of $n = 1$, of course $\chi(T) = 1$.

c) Note that a full m-ary tree with n nodes has the maximum height if at each level of the tree, only 1 node has children and all the others at that level are leaves (since this way we maximize the depth we can go while still obeying the rules of full m-ary trees). Such a tree can be shown as:



I claim that for such a tree with maximized height, it's number of nodes can be written as $n = l + i = l + h = m + (m-1)(h-1) + h$ where h is the height, l is the number of leaves and i is the number of internal nodes (note that for such a tree, the number of internal nodes is equal to its height, since at each level from 0 to $h-1$ only 1 internal node exists at that level). To prove this claim I will use induction on the tree height.

The base case is where $h = 1$. For such a m-ary tree it is clear to see that the number of nodes is $n = m + 1 = m + (m-1)(h-1) + h$.

Then for the inductive step, let's assume this holds for $h - 1$ such that $n_{h-1} = m + (m-1)(h-2) + (h-1)$. Then for a tree with height h, it's number of nodes is, the number of nodes at the previous level $+m$, since at each level the only addition comes from 1 internal node's children:

$$n_h = m + n_{h-1} = m + m + (m-1)(h-2) + (h-1) = m + (m-1)(h-1) + h$$

So, I have proven my claim for the number of nodes in a tree with maximum height. Then going back from this, we can find the maximum height for a tree when we are given n and m such that:

$$n = m + (m-1)(h-1) + h = mh + 1 \text{ where this means } h = \frac{n-1}{m}$$

The upper bound on the height of a full m-ary tree with n vertices in terms of m and n is $h = \dfrac{n-1}{m}$.