Homework 4

MATH 263: Discrete Mathematics 2

Dr. Petrescu

Due: March 31, 2023 **Denny Cao**

Question 1.

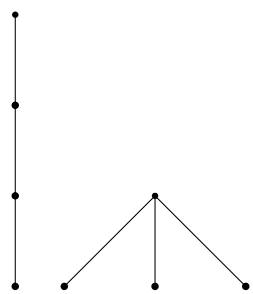
(i) How many nonisomorphic non rooted trees are there with 4 vertices?

(ii) How many nonisomorphic rooted trees are there with 4 vertices?

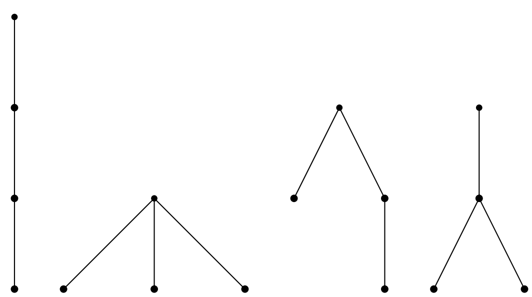
(iii) How many nonisomorphic **non rooted** trees are there with 5 vertices?

Answer 1.

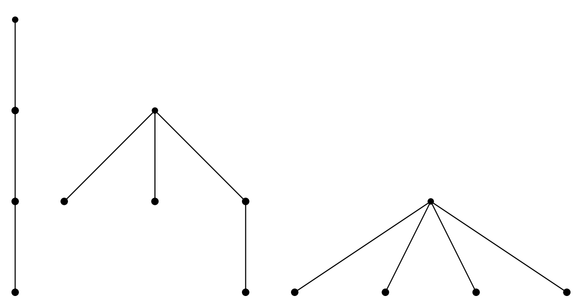
(i) 2



(ii) 4



(iii) 3



Question 2.

a) How many edges does a tree with 10,000 vertices have?

b) How many vertices does a full 5-ary tree with 100 internal vertices have?

c) How many edges does a full binary tree wtih 1,000 internal vertices have?

d) How many leaves does a full 3-ary tree with 100 vertices have?

Answer 2.

a) A tree with n vertices has $n-1$ edges. Thus, a tree with 10,000 vertices has $10,000-1 = 9,999$ edges.

b) A full m-ary tree with i internal vertices has $n = mi+1$ vertices. Thus, a full 5-ary tree with 100 internal vertices has $n = 5 \times 100 + 1 = 501$ vertices.

c) A binary tree is a full 2-ary tree. Thus,a full binary tree wtih 1,000 internal vertices has $n = 2 \times 1,000 + 1 = 2,001$ vertices.

d) A full m-ary tree with n vertices has $\ell = \frac{(m-1)n+1}{m}$ leaves. Thus, a full 3-ary tree with 100 vertices has $\ell = \frac{(3-1)\times 100+1}{3} = 67$ leaves.

Question 3. Suppose that the address of the vertex v in the ordered rooted tree T is 3.4.5.2.4.

a) At what level is v?

b) What is the address of the parent of v?

c) What is the least number of siblings v can have?

d) What is the smallest possible number of vertices in T if v has this address?

Answer 3.

a) 5

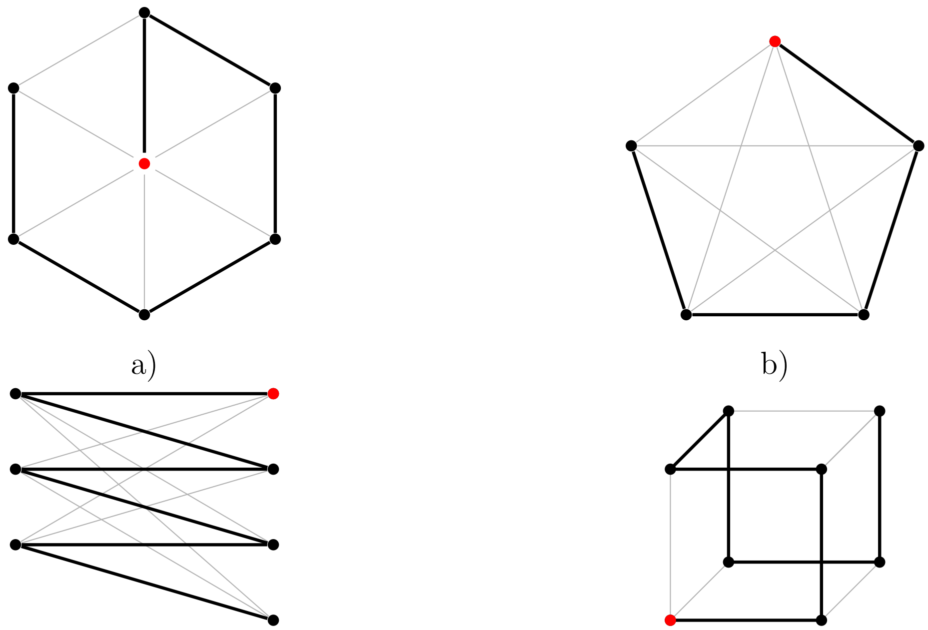
b) 3.4.5.2

c) 3

d) 15

Question 4. Use depth- first search to find a spanning tree of each of these graphs.
a) W_6 , starting at the vertex of degree 6 b) K_5
c) $K_{3,4}$, starting at a vertex of degree 3 d) Q_3

Answer 4.



a) b) c) d)

Question 5. Prove Kruskal's Theorem.

Answer 5.

Theorem 1. Kruskal's Algorithm produces a minimal spanning tree of a connected simple graph.

Proof. The proof consists of two parts. The first part is to show that the algorithm produces a spanning tree. The second part is to show that the algorithm produces a minimal spanning tree.

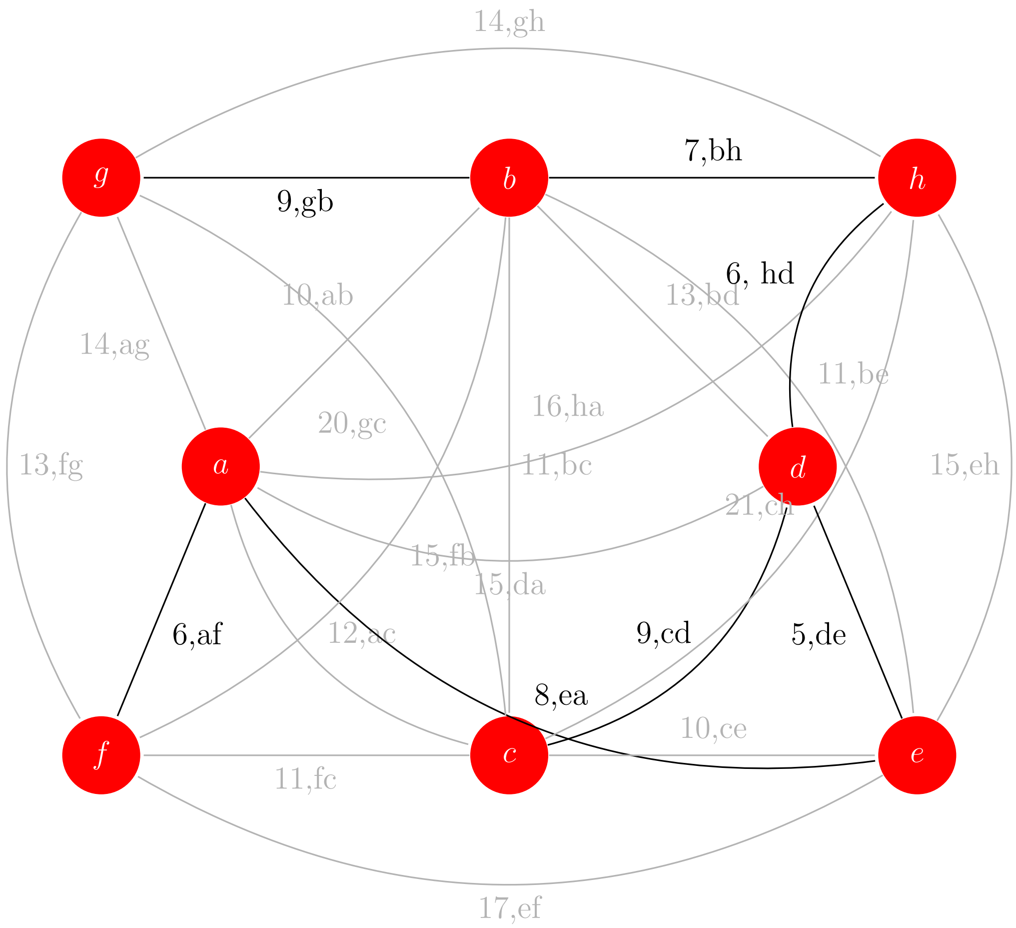
Kruskal's Algorithm produces a spanning tree. A spanning tree is a tree that is connected and contains no cycles. Let S be the subgraph of a connected simple graph G which is the output of Kruskal's Algorithm. S must be connected because otherwise, the algorithm would not have terminated. S must also contain no cycles because each time an edge is added to S, the algorithm checks to see if the edge creates a cycle. If the edge creates a cycle, the edge is not added to S. Therefore, S is a spanning tree.

The spanning tree produced by Krushal's Algorithm is minimal. We will prove by contradiction. Assume for purposes of contradiction that S is not minimal. Then, there must be an MST with edges not in S. Let T be the MST with the least number of edges.
Consider the first edge added to S by Krushal's Algorithm but not to T, e. Then, $T \cup \{e\}$ must contain a circuit, and, since S has no circuits, one of the edges in the circuit must not be in S. Let this edge be c. Then, a graph $T' = (T \cup \{e\}) \setminus \{c\}$ is a spanning tree of G. As T is an MST, $w(e) \geq w(c)$. Also, as e was selected by Kruskal's Algorithm, it must be of minimal weight, meaning $w(e) \leq w(c)$, which implies $w(e) = w(c)$. Therefore, T' is also an

MST of G, but it has one more edge in common with S than T. ※

We arrive as a contradiction, as T has the most edges in common with S of any MST of G. Therefore, S is a minimal spanning tree. □

Question 6. Use Prim-Jarnik's or Kruskal's algorithm to find, step by step, the minimal spanning tree from the graph below. State what method you are using.



Answer 6. Using Kruskal's Algorithm, we start with the edge of least weight. We add this edge to the spanning tree. We then continue to add edges of least weight, but only if they do not create a circuit. Each iteration of the algorithm is shown below:

1. de
2. hd
3. af
4. bh
5. ea

6. cd

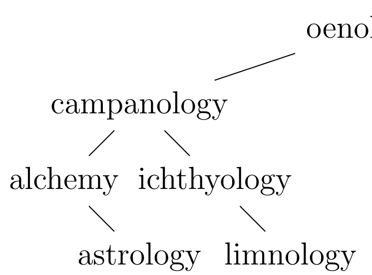
7. gb

Question 7. Describe the tree produced by breadth-first search and depth-first search for the n-cube graph Q_n, where n is a positive integer.

Answer 7.

Question 8. Build a binary search tree for the words: *oenology, phrenology, campanology, ornithology, ichthyology, limnology, alchemy,* and *astrology* using alphabetical order.

Answer 8.



Question 9. For the tree in Question 8 determine the order in which a inorder traversal visits the vertices of the given ordered rooted tree.

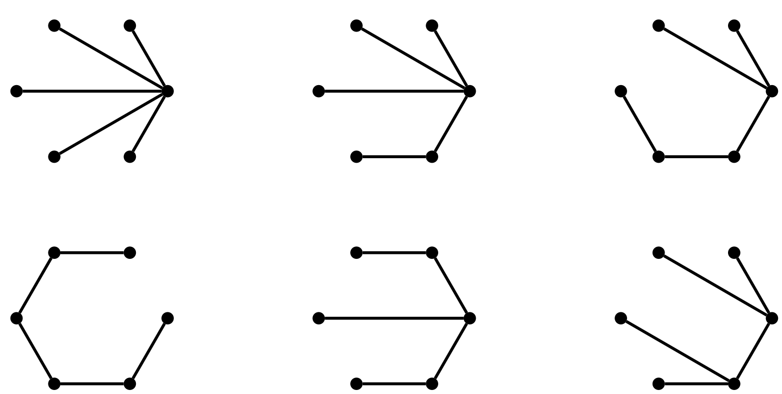
Answer 9. astrology, alchemy, campanology, limnology, ichthyology, oenology, ornithology, phrenology

Question 10. For the tree in Question 8 determine the order in which a postorder traversal visits the vertices of the given ordered rooted tree.

Answer 10. astrology, alchemy, limnology, ichthyology, campanology, ornithology, phrenology, oenology

Question 11. How many nonisomorphic unrooted trees are there with six vertices?

Answer 11. There are 6 nonisomorphic unrooted trees with six vertices.

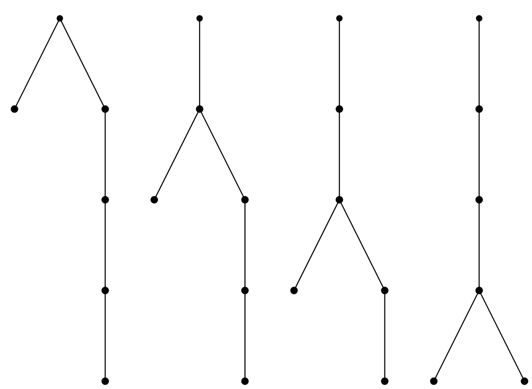


Question 12. How many nonisomorphic rooted trees are there with six vertices?

Answer 12. There are 20 nonisomorphic rooted trees with six vertices.

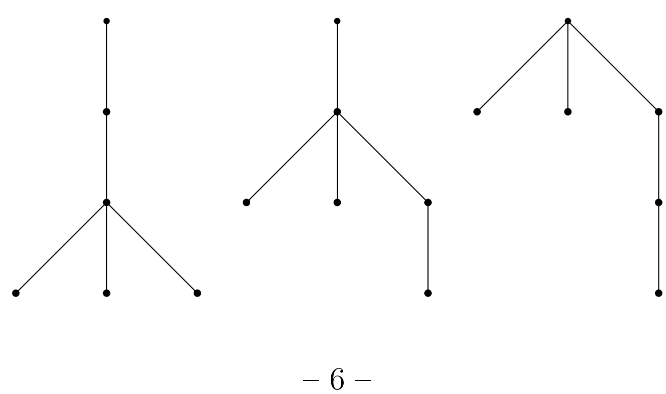
There is only 1 rooted tree with 6 vertices with a path of length 5.

With a path of length 4, there are 4 possibilities as we can attach the last leaf of the rooted tree with a path of length 5 to one of the top four vertices.

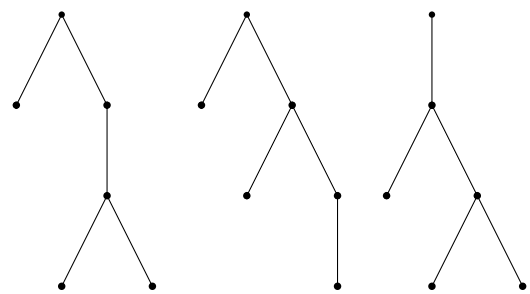


With a path of length 3, there are 8 possibilities.

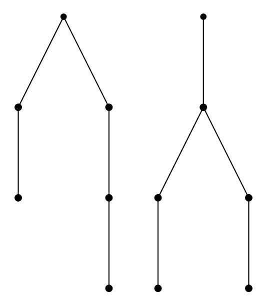
We can attach 2 leaves to the first 3 vertices (not the last, as that would create a path of length 4).



We can also choose 2 places from 3 vertices to place the last 2 leaves, giving us another 3 possibilities.

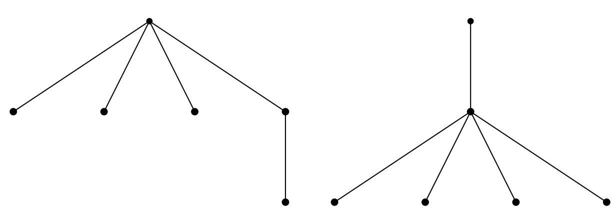


We can also attach the last two vertices in a path of length 2 to one of the top two vertices (we cannot attach to the bottom 2, as that would create a path greater than 3).

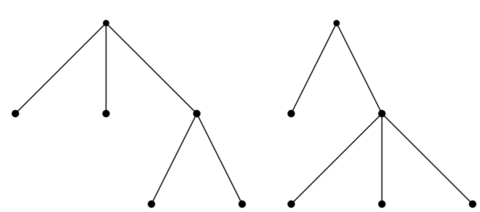


With a path of length 2, there are 6 possibilities.

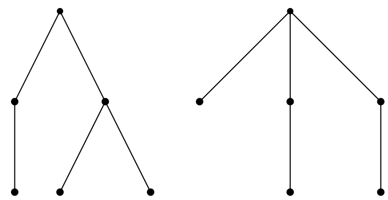
We can place 3 leaves on the first 2 vertices (not the last, as that would create a path of length 3).



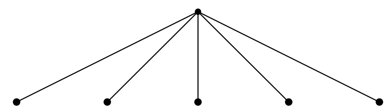
We can also attach 2 leaves to the root and 1 to the second vertex, or 1 to the root and 2 to the second vertex.



We can also attach the last two vertices as a path of length 1 to the root, leaving the last leaf to be attached to a root at level 1 or to the root of the tree.



With a path of length 1, there is 1 possibility.



Altogether, there are $1 + 4 + 8 + 6 + 1 = 20$ non-isomorphic trees with 6 vertices.