Scott Fenton

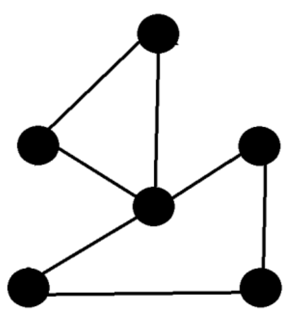
Homework 6

CS320L

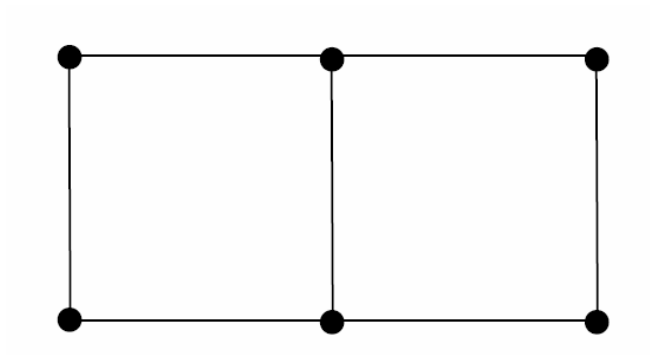
Question 1: Find a Nonisomorphic Graph

a) Draw another graph H that has the same number of vertices and edges and the same degrees as G but is not isomorphic to G.

Graph H



Graph G



b) Is there another invariant we discussed besides the number of vertices and edges and the degrees, such as the length of circuits and paths, that could be used to show that G and H are nonisomorphic? If so, please state how this invariant differs between the two graphs.

Yes, the circuit length between graph H and graph G are different. Graph H has circuit length of 3, while graph G has a circuit of length 4.

Question 2: … And More about Graphs

a) In class we showed that the cycle C_6 is bipartite. Which of the cycles C_3, C_4, C_5, and C_7 are bipartite as well?

C_4 is also bipartite.

b) **Bonus:** Based on (a) there seems to be a rule that tells us for which numbers n the cycle C_n is bipartite. What is that rule? Give an argument for this rule to be correct for all integers $n > 2$.

The rule is that any even integer of n, which is greater than 2, the graph will be bipartite. This rule holds true because any case where n is odd, will result in a graph with two adjacent vertices of the same color. While an even number of vertex will result in the two adjacent vertex to the last vertex being the same color.

c) How many nonisomorphic subgraphs does C_4 have? Draw them.

Insert Images

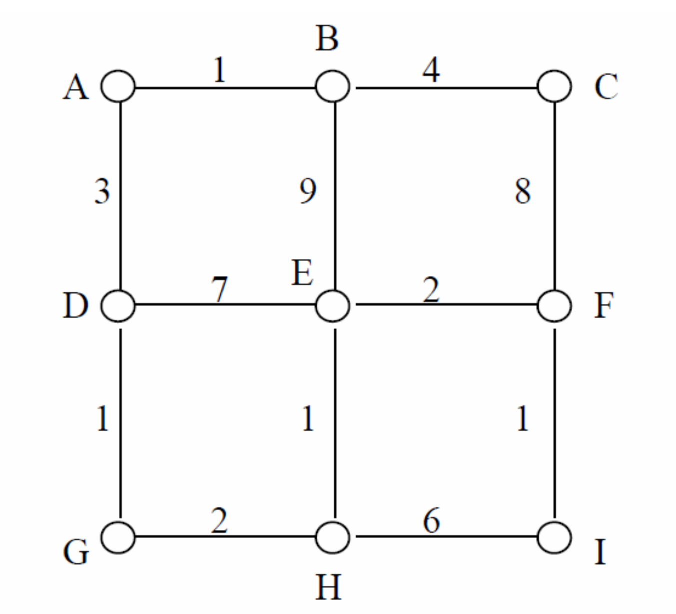
d) Develop an equation for computing the number of edges $|E_n|$ for any wheel W_n. Explain your reasoning and test your equation by computing $|E_3|$.

$E_n = 2(n – 1)$
$E_3 = 2(3 – 1)$
$E_3 = 2(2)$
$E_3 = 4$

This equation makes sense because for a wheel with vertices 4, there are 3 edges from the cycle of a graph 3, and 3 edges connecting each vertex from the cycle 3 graph to the center vertex. This pattern continues, for each wheel graph, the base graph ignoring the center vertex, creates n-1 edges, and the center vertex, creates n -1 edges.

Question 3: Dijkstra in Action

Use Dijkstra's algorithm to compute the shortest path from A to I, where edge labels indicate the distance (or cost) between vertices. For each iteration, write down the shortest path (i.e., its length and the vertices in it) from A to each vertex that has been found so far, and also indicate which vertices are currently in the set S.



Vertices in set S are shown in ***Italics and Bold***.

0.) A: 0; B:∞; C: ∞; D: ∞; E: ∞; F: ∞; G: ∞; H: ∞; I: ∞
1.) ***A: 0***; B:1(A); C: ∞; D: 3(A); E: ∞; F: ∞; G: ∞; H: ∞; I: ∞
2.) ***A: 0; B:1(A);*** C: 5(A,B); D: 3(A); E: 10(A,B); F: ∞; G: ∞; H: ∞; I: ∞
3.) ***A: 0; B:1(A);*** **C: 5(A,B);** D: 3(A); E: 10(A,B); F: 12(A,B,C); G: ∞; H: ∞; I: ∞
4.) ***A: 0; B:1(A); C: 5(A,B); D: 3(A);*** E: 10(A,B); F: 12(A,B,C); G: 4(A,D); H: ∞; I: ∞
5.) ***A: 0; B:1(A);*** **C: 5(A,B); D: 3(A);** E: 10(A,B); F: 12(A,B,C); **G: 4(A,D);** H: 6(A,D,G); I: ∞
6.) ***A: 0; B:1(A);*** **C: 5(A,B); D: 3(A);** E: 7(A,D,G,H); F: 12(A,B,C); **G: 4(A,D); H: 6(A,D,G);** I: 12(A,D,G,H)
7.) ***A: 0; B:1(A);*** **C: 5(A,B); D: 3(A);** ***E: 7(A,D,G,H);*** F: 7(A,D,G,H,E); **G: 4(A,D); H: 6(A,D,G);** I: 12(A,D,G,H)
8.) ***A: 0; B:1(A);*** **C: 5(A,B); D: 3(A);** ***E: 7(A,D,G,H); F: 7(A,D,G,H,E);*** **G: 4(A,D); H: 6(A,D,G);** I: 10(*A,D,G,H,E,F*)
9.) ***A: 0; B:1(A);*** **C: 5(A,B); D: 3(A);** ***E: 7(A,D,G,H); F: 7(A,D,G,H,E);*** **G: 4(A,D); H: 6(A,D,G); I: 10(*A,D,G,H,E,F*)**

The shortest path from A to I, passes through the vertices, A-D-G-H-E-F-I, with a total distance of 10.

Question 4: Trees

a.) How many vertices does a full 4-ary tree with 100 internal vertices have?

V = m*i + 1

V= 4*100 + 1

V = 401

The tree has 401 vertices.

b.) How many vertices and how many leaves does a complete *m*-ary tree of height *h* have?

Vertices:

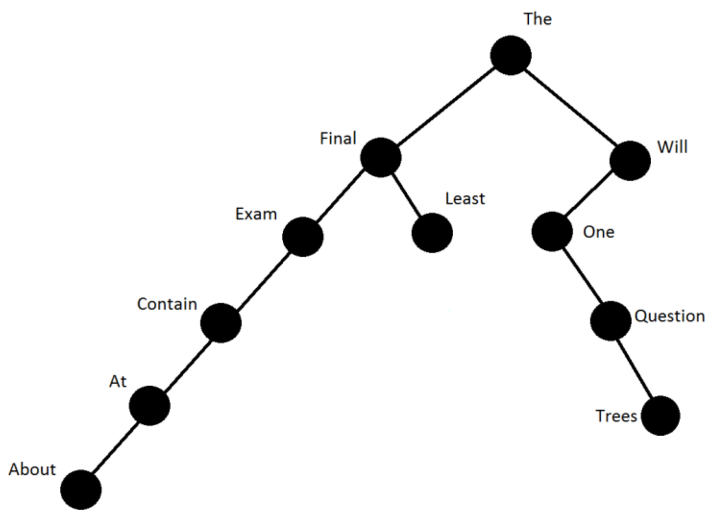
$$\sum_{n=0}^{h} m^n = \frac{(m^{h+1} - 1)}{(m-1)}$$

Where we sum the vertices m^n at each depth-level until we hit the root node(height).

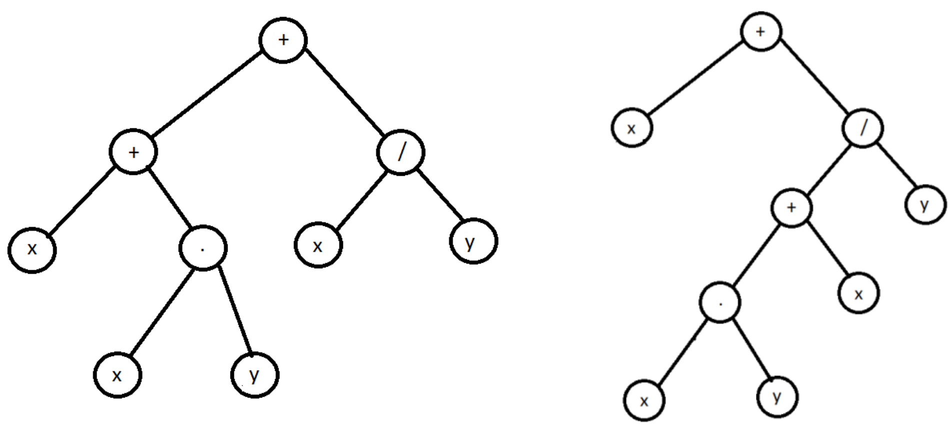
Leaves:

We can have m^h leaves, where m is the total number of children nodes.

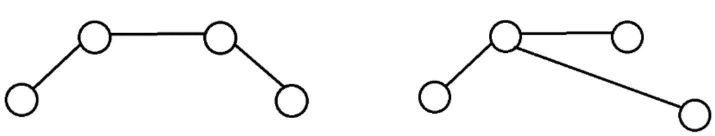
c.) Build a binary search tree for the words *the, final, exam, will, contain, at, least, one, question, about,* and *trees* using alphabetical order and adding words in the same order as listed here.



d.) Represent the expressions $(x + x \cdot y) + (x/y)$ and $x + ((x \cdot y + x)/y)$ using two binary trees.



e.) How many non-isomorphic trees with four vertices are there? Draw them.



f) **Bonus:** Show that a full m-ary tree with l leaves has $(l - 1)/(m - 1)$ internal vertices.

The total number v of vertices in a tree is the sum of the internal vertices I, and the leaves l:

V = i + L

We already know that for a full m-ary tree it is true that

V = mi + 1

<u>Then it follows that:</u>

1.) i + L = mi + 1
2.) L = mi + 1 – i
3.) (L – 1) = mi – i
4.) (L – 1) = i(m – 1)
5.) **i = (L – 1) / (m – 1), where (i) is internal vertices.**