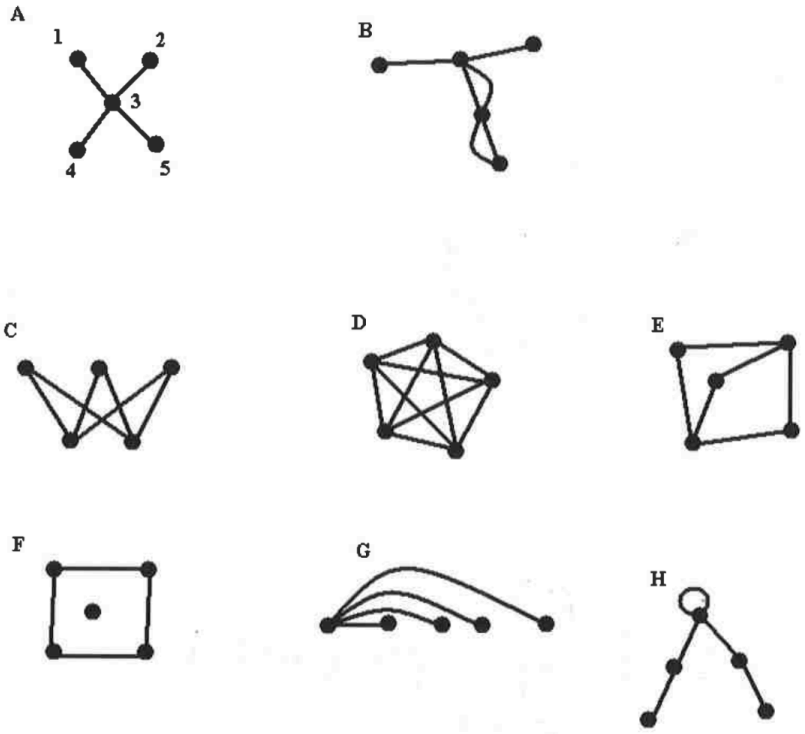
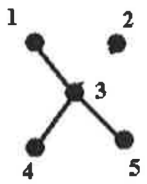
1. Please answer questions 1 - 7 using the following graphs.



1. Which of the graphs have loops?
2. Which of the graphs are simple?
3. Which graphs are complete?
4. Which graphs are connected?
5. Which graphs are acyclic?
6. Which graphs are bipartite complete?
7. Which paris of graphs are isomorphic?

In [ ]:

2. Please answer questions 8 - 9 using the following graph.

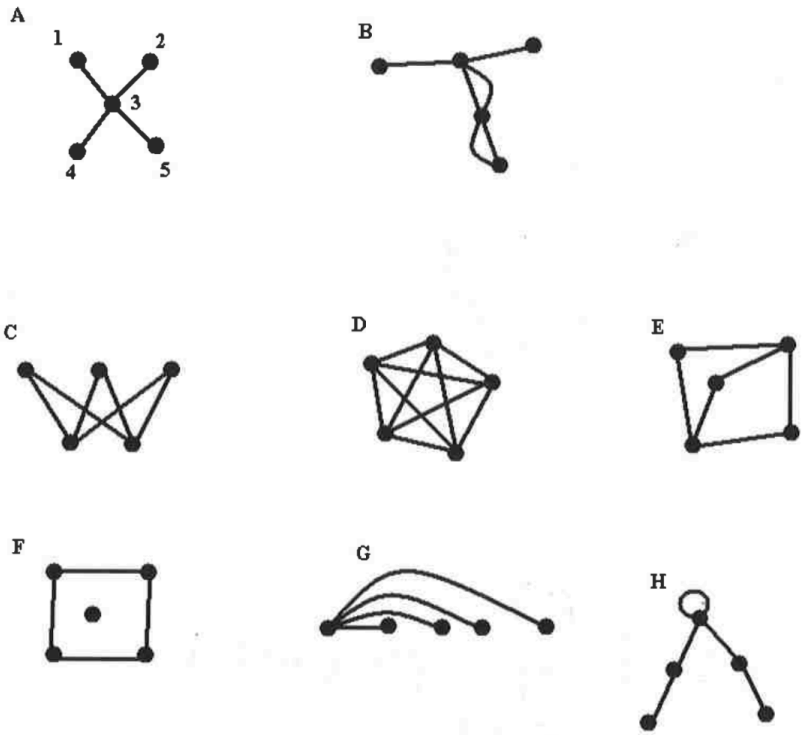


8. Give the adjacency matrix representation for the graph.

9. Give the adjacency list representation for the graph.

In [ ]:

3. Please answer questions 1 - 7 using the following graph.



1. Which of the above are trees?
2. Draw a complete binary tree that is not full with at least 3 nodes. Number the nodes.

3. What is the depth of the binary tree you draw?
4. Name a leaf.
5. What is the root?

6. Give the pointer representation for your tree of #2.

7. Give the left child-right child array representation for your tree of #2.

In [ ]:

4.1. Please draw the algebraic expression tree for the expression: $(4x + 7y) * (3z - 2x)$

4.2. Please traverse the algebraic expression tree in 4.1. using prefix and postfix algorithm

In [ ]:

5.1. Any algorithm that solves the search problem for a 957 element list by comparing the target element x to the list items must do at least ___ comparisons.

5.2. Any algorithm that sorts a 6 element list by comparing pairs of items from the list must do at least ____ comparisions in the worst case.

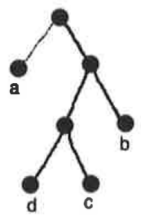
In [ ]:

6. Given the following code, decode the received bit string:

1110100110

a	b	c	d	e
100	111	101	110	0

7. Given the following tree, give the code words for the characters: a, b, c, and d.



8. Construct the huffman tree for the following characters and their associated frequencies.

character	a	b	c	d
frequency	35	11	45	9

1. Please construct the huffman tree for the above characters and associated frequencies.

2. Please write the code words for the characters a, b, c, and d.

3. Please encode the word **cab** using the huffman code.

4. If the file containing **100** characters consisted of the above characters and associated freuencies. How many bits would be needed to store using the associated huffman code?

In [ ]: