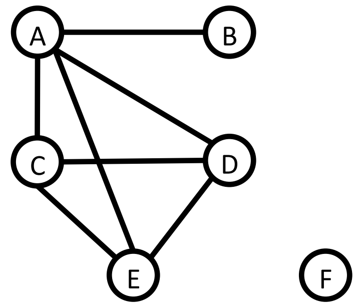
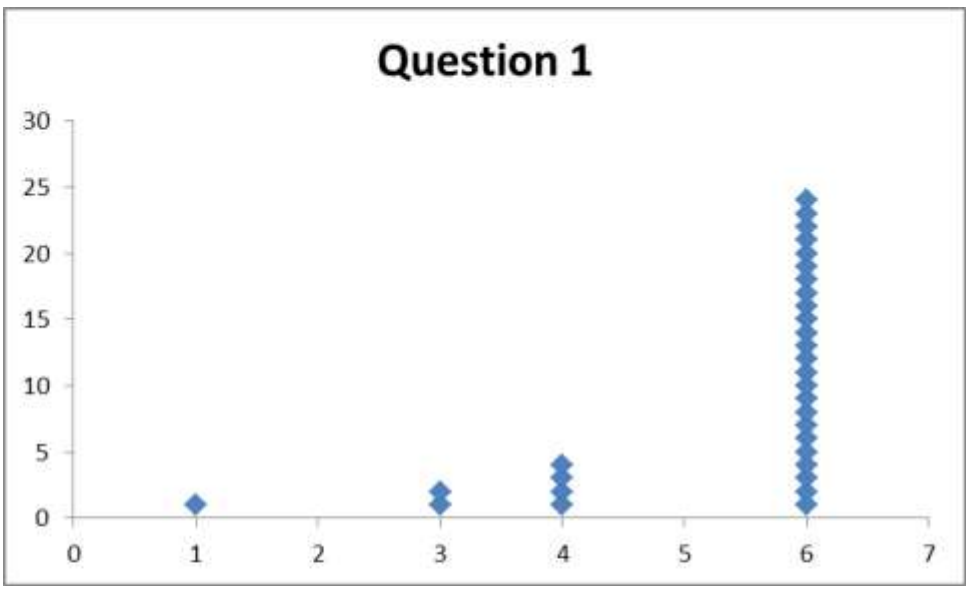
Name ____ Discrete II, Test 2, Spring 2017

1) Find an incidence matrix of the graph shown below. (6 points)

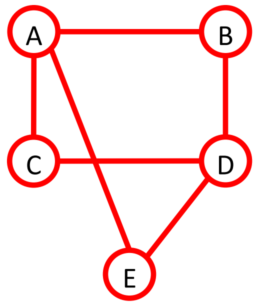
$$\begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

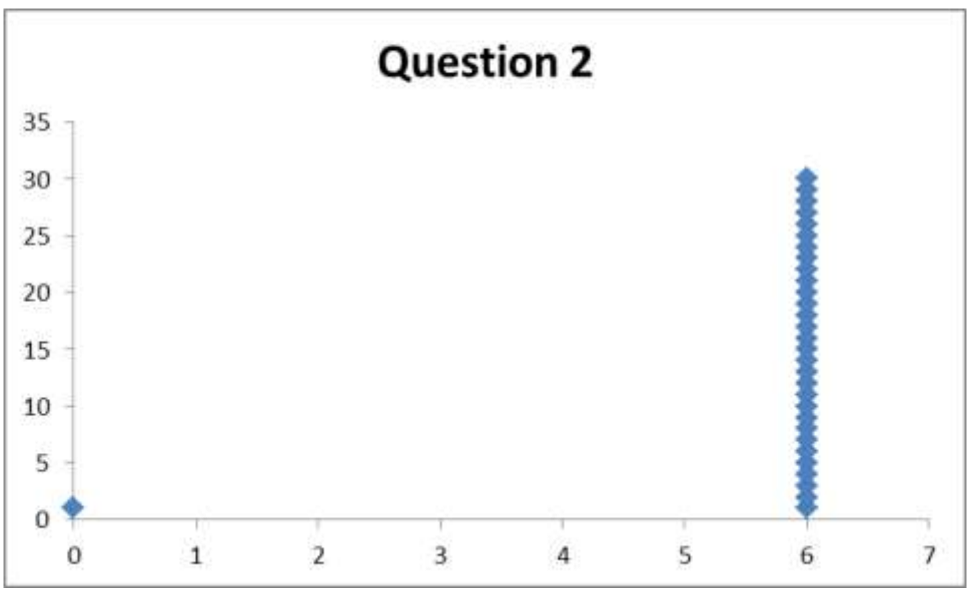




2) Draw a graph with the adjacency matrix below. (6 points)

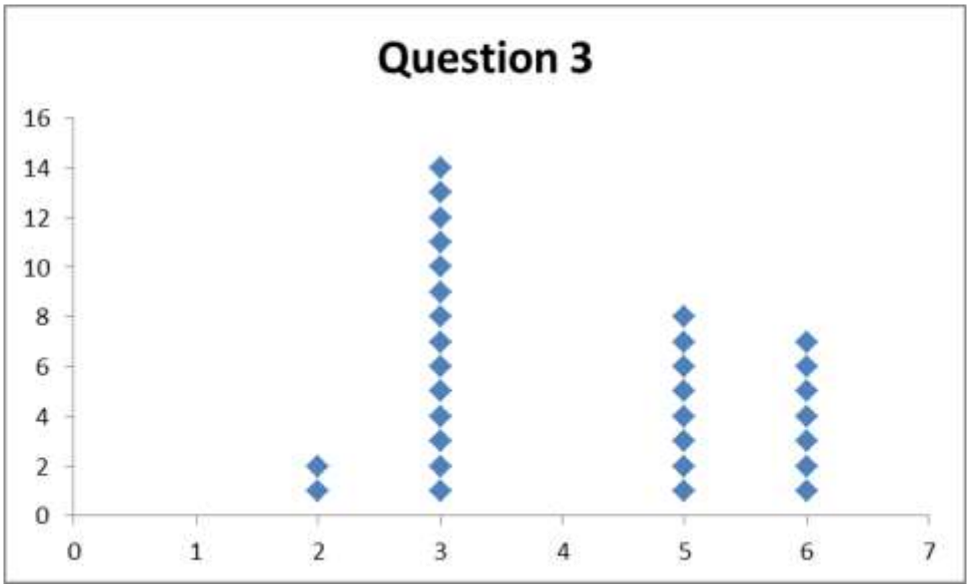
$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$





3) It is known that the number of an edges in a planar graph with n vertices is bounded by $O(n)$. We also saw that the runtime for Dijkstra's algorithm was bounded by $O(n\log(n) + m)$. Use these facts to find the a bound for the runtime for Dijkstra's algorithm on a planar graph. (6 points)

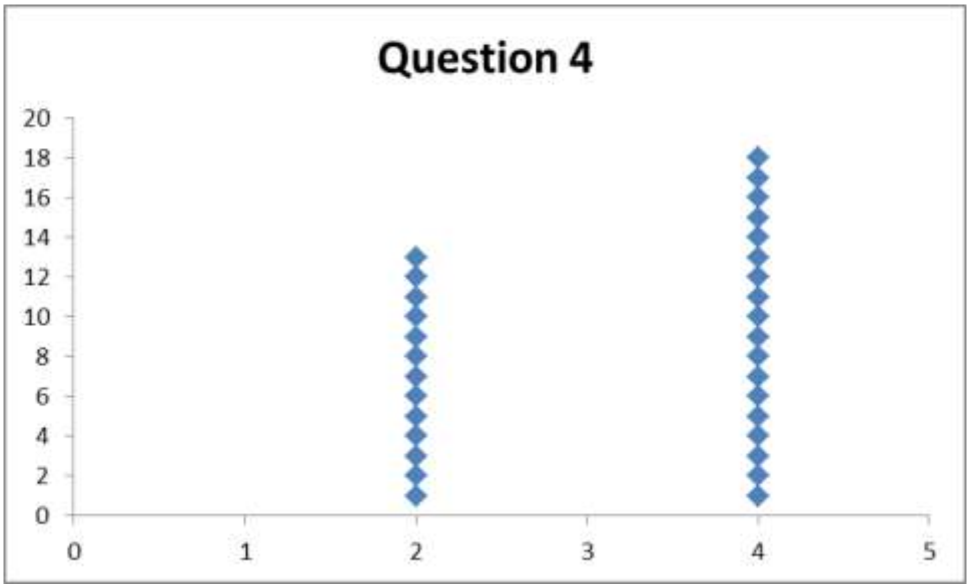
$O(n\log(n) + O(n))$ which is $O(n\log(n))$



In the **five** problems that follow, use n for the number of vertices and m for the number of edges in a graph.

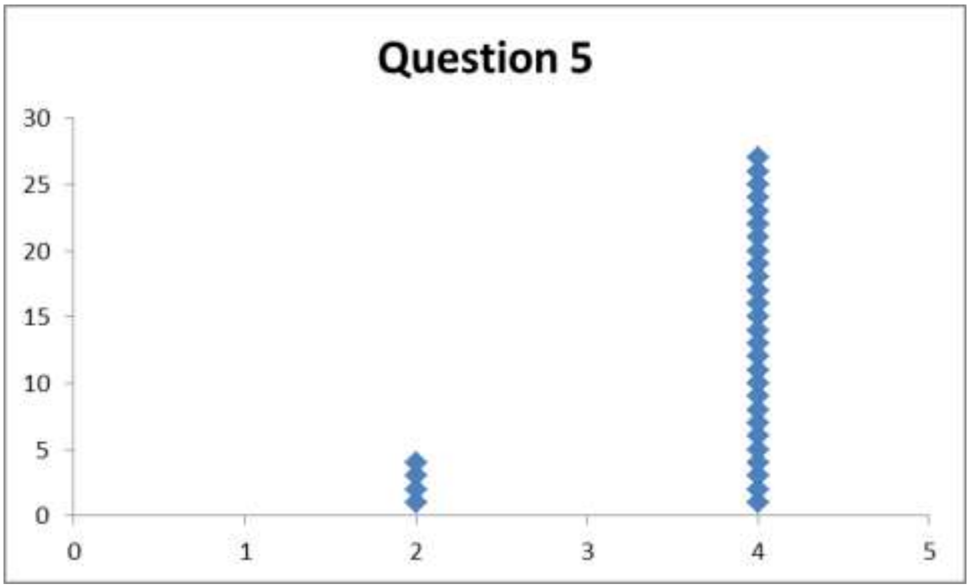
4) Find an upper bound, using big-Oh, on total size of an incidence matrix. (4 points)

$O(nm)$



5) Find an upper bound, using big-Oh, on the total size of an adjacency matrix. (4 points)

$O(n^2)$

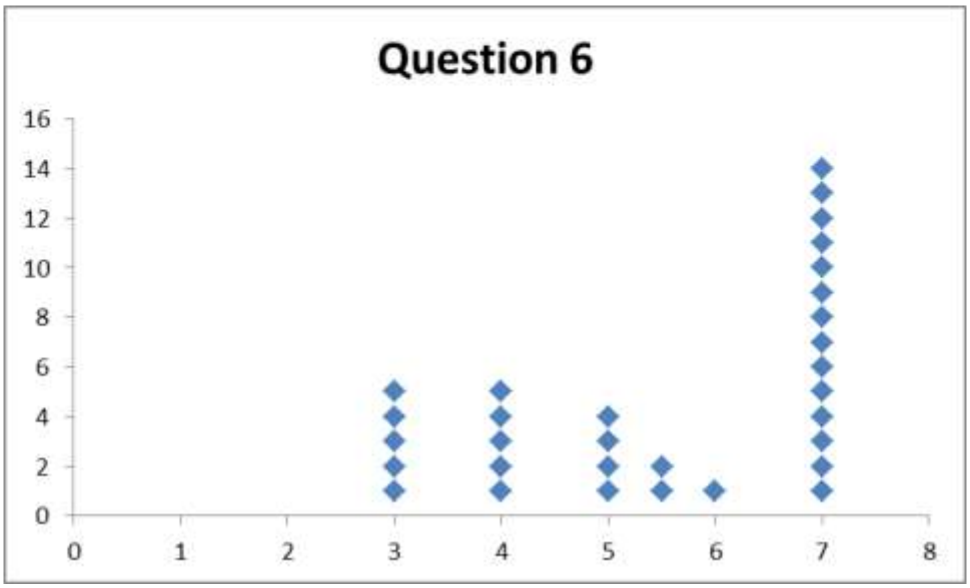


6) Consider the problem "Find the shortest cycle through vertex A". Find and justify a nonconstant lower bound, using big-Omega, for the time required to solve this problem. (7 points)

$\Omega(n)$ – because the cycle might go through all vertices.

$\Omega(m)$ – because the edges might be ordered such that the ones in the cycle come last and so you end up looking at every single edge.

$\Omega(n+m)$ because of the two bounds above.



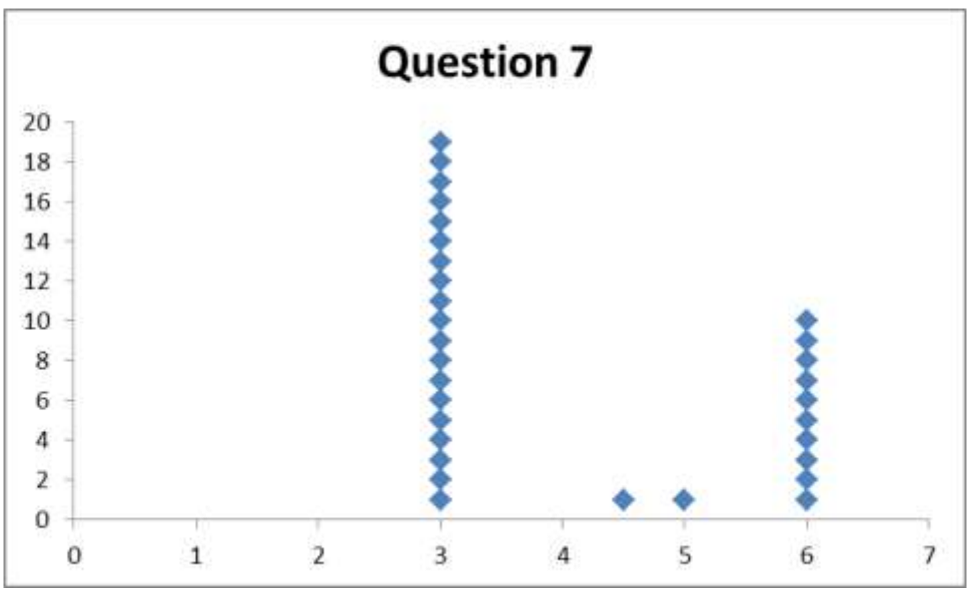
7) Consider the problem "Find the shortest cycle through vertex A". It can be solved by creating a spanning tree in a breadth-first manner. (Create the tree, but at each vertex you add to the tree, check to see if it is adjacent to A). Specifically we can implement this algorithm with an incidence matrix as follows:

Starting from vertex A, search through row A to find all vertices of level 1.
For each vertex of level 1, search through their row to find all vertices of level 2.
Repeat for vertices of level 2, 3, 4, etc.

Use this algorithm to find an upper bound, using big-Oh, for the time required to solve this problem. (6 points)

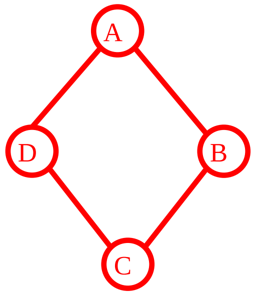
On the incidence matrix, there are m items in each row to look at and we iterate through all n rows. Each column will be considered exactly twice as we iterate through the rows. Everything else is constant time, so we get the bound:

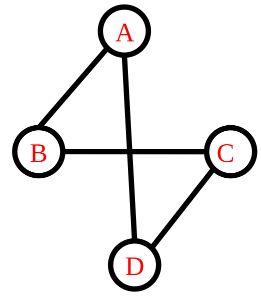
$$O(nm + 2nm) = O(nm)$$

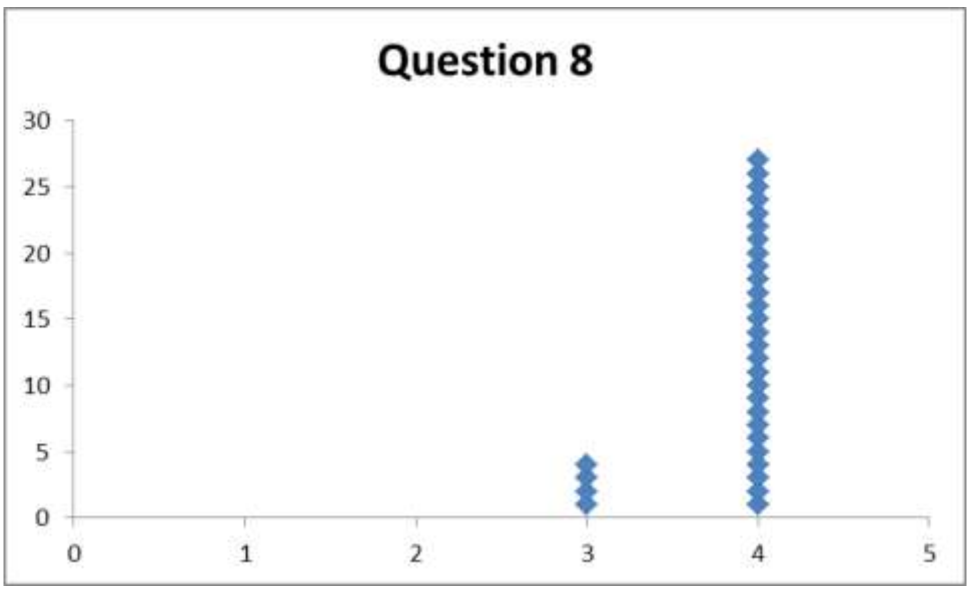


8) Is the graph below a cycle? Justify your answer by redrawing the graph. (4 points)

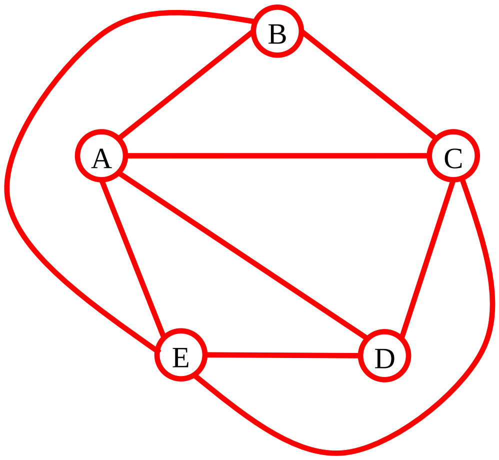
Yes it is:

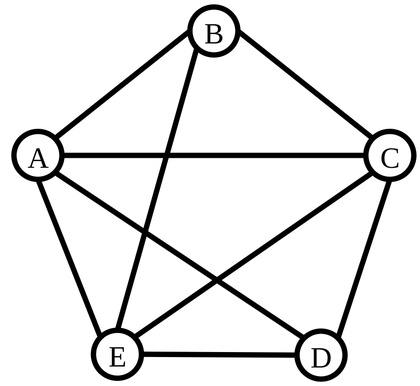


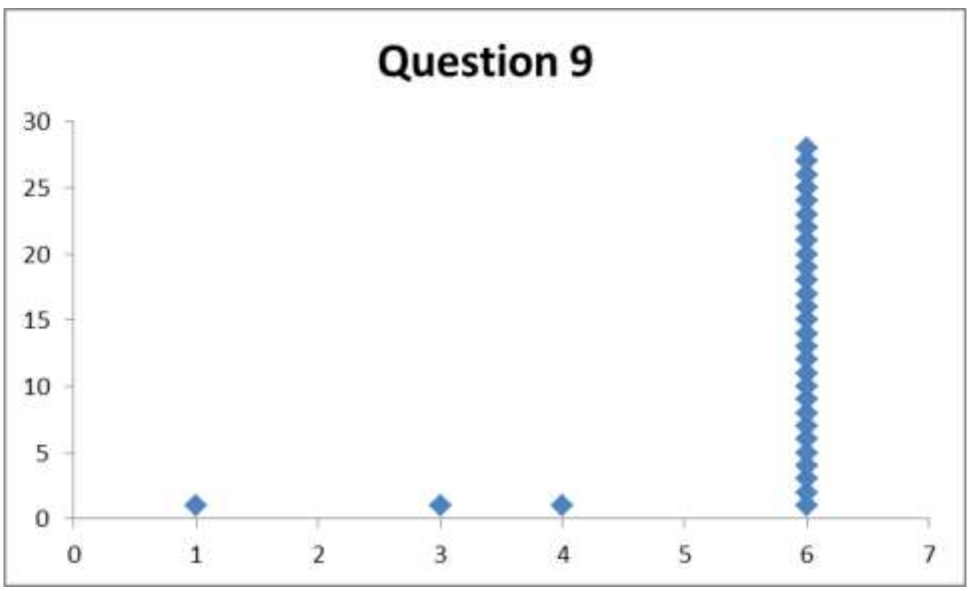




9) The graph below is planar. However, the drawing of it is not a planar representation. Draw a planar representation of this graph. (6 points)

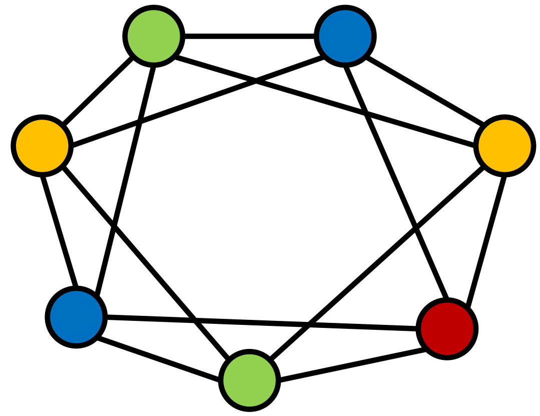


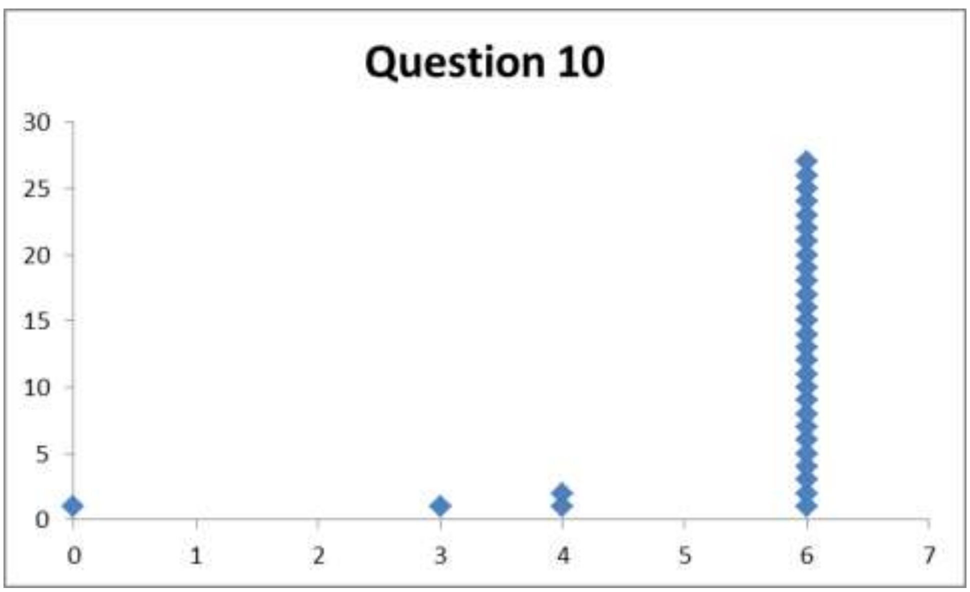




10) Color the graph below with as few colors as possible.

(Recall that adjacent vertices cannot share a color) (6 points)



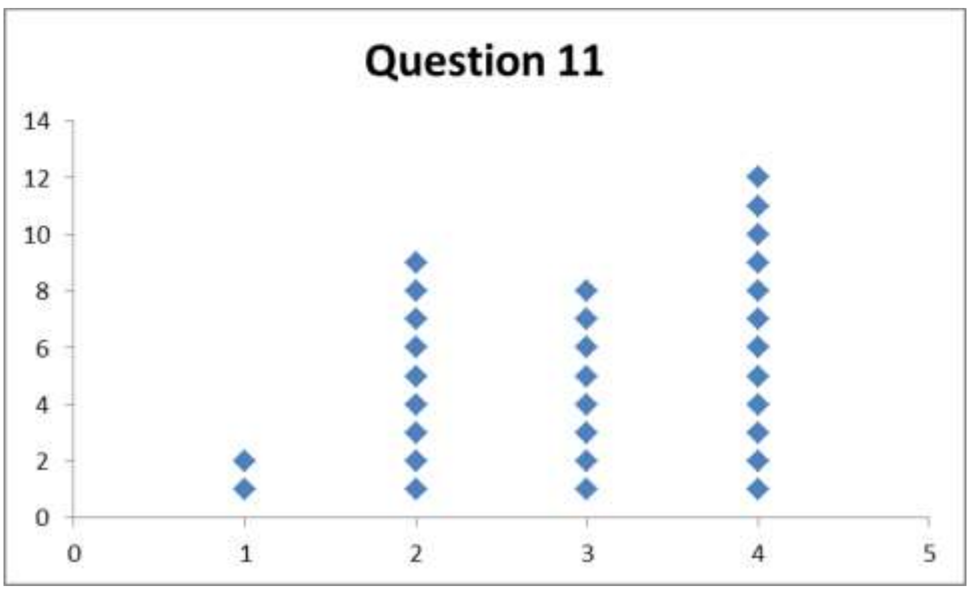


11) Suppose a graph has n vertices, m edges, and every vertex has degree at most D. Consider the coloring algorithm that sequentially colors each vertex with any valid color. If you reuse colors whenever possible express an upper bound, using big-Oh, on the number of colors required by this algorithm. (4 points)

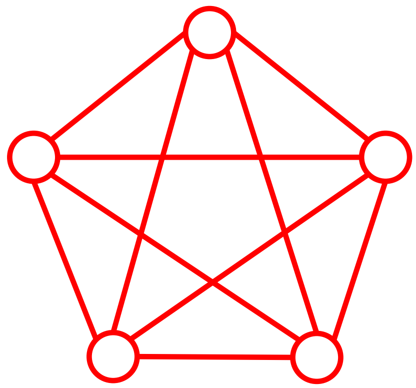
The exact answer here is $D + 1$ because if each vertex has at most D neighbors, we can eliminate each of their colors. With $D + 1$ color options, there is always another color to use. The asymptotics simplify this a little bit to:

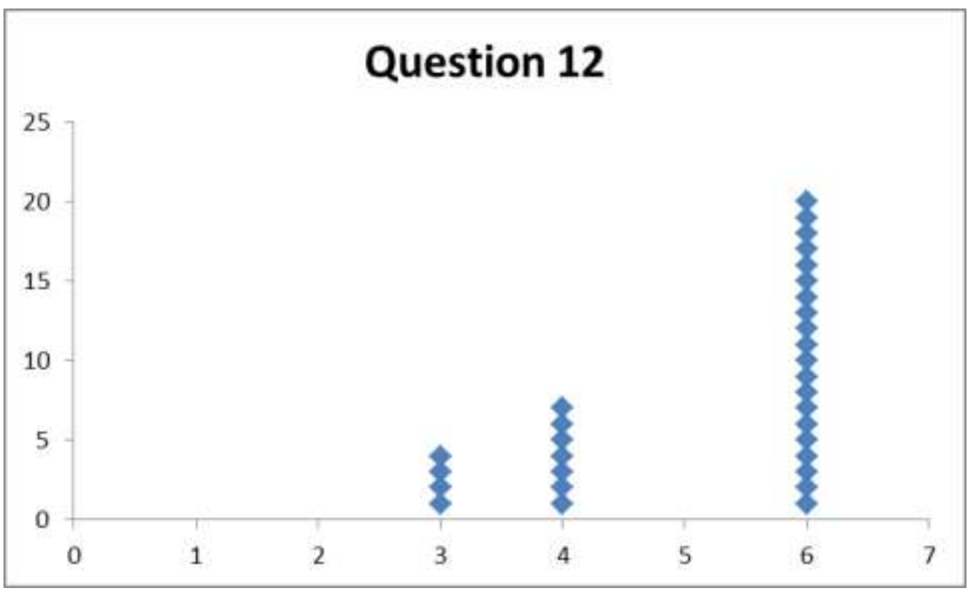
$$O(D)$$

(Note: I had wanted you to figure out that the bound is actually $D + 1$, but realized too late that I actually asked for big-Oh)

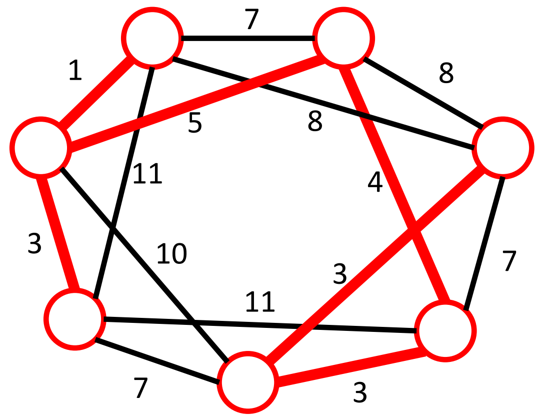


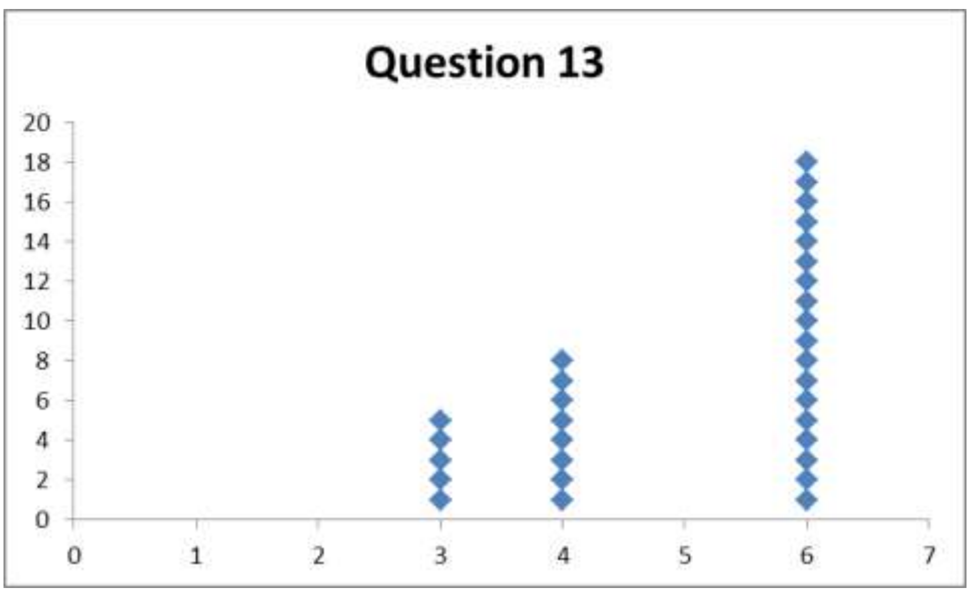
12) Draw the complete graph K_5. (6 points)





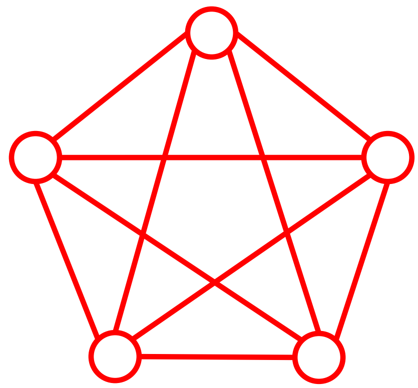
13) Find a minimal spanning tree of the graph shown below. (6 points)

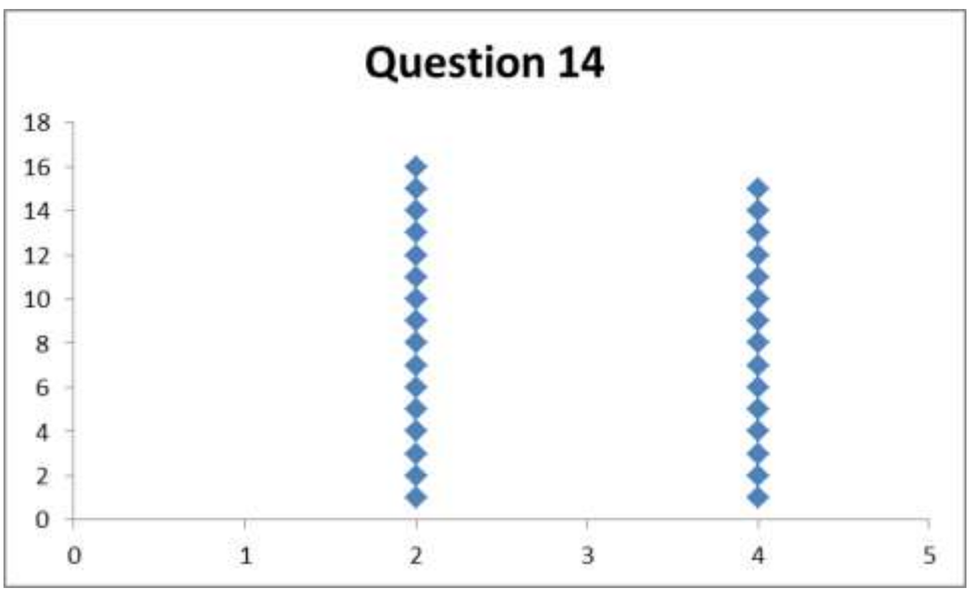




14) Draw a graph that is not planar. (4 points)

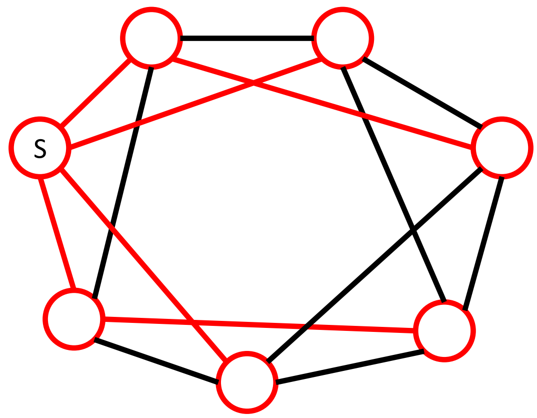
($K_{3,3}$ or something that contains something with a structure similar to one of these also works)

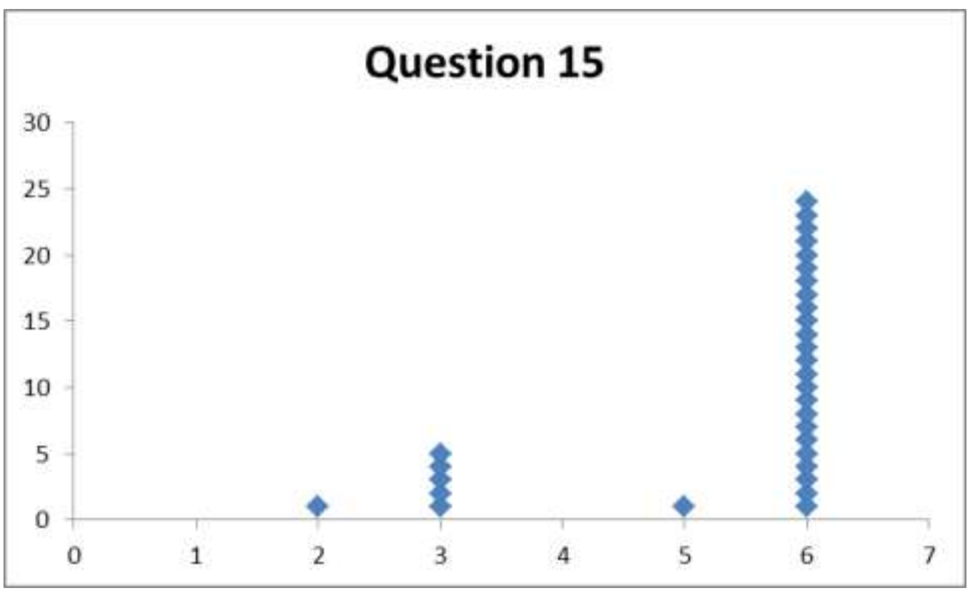




15) Use a breadth-first algorithm to find a spanning tree of the graph shown below. Start at the vertex labelled “S”. (6 points)

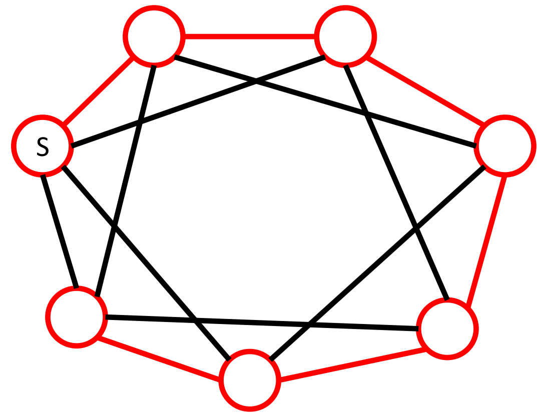
There are multiple correct answers.

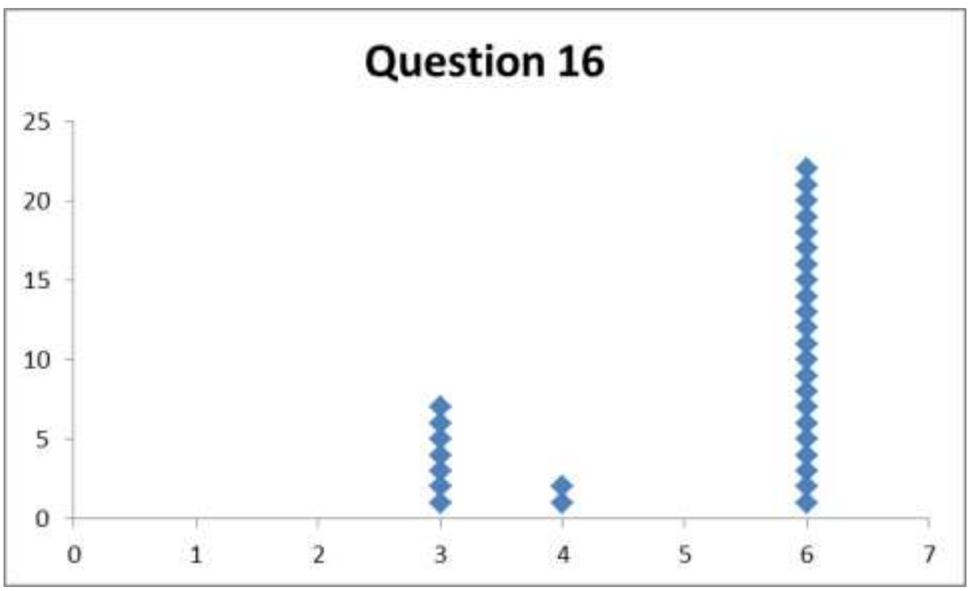




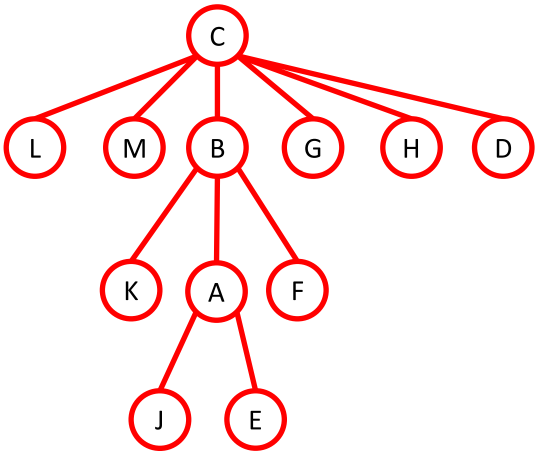
16) Use a depth-first algorithm to find a spanning tree of the graph shown below. Start at the vertex labelled “S”. (6 points)

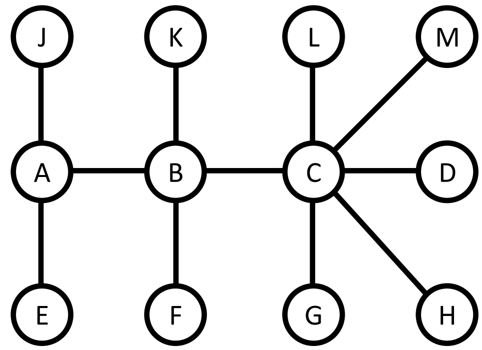
There are multiple correct answers.

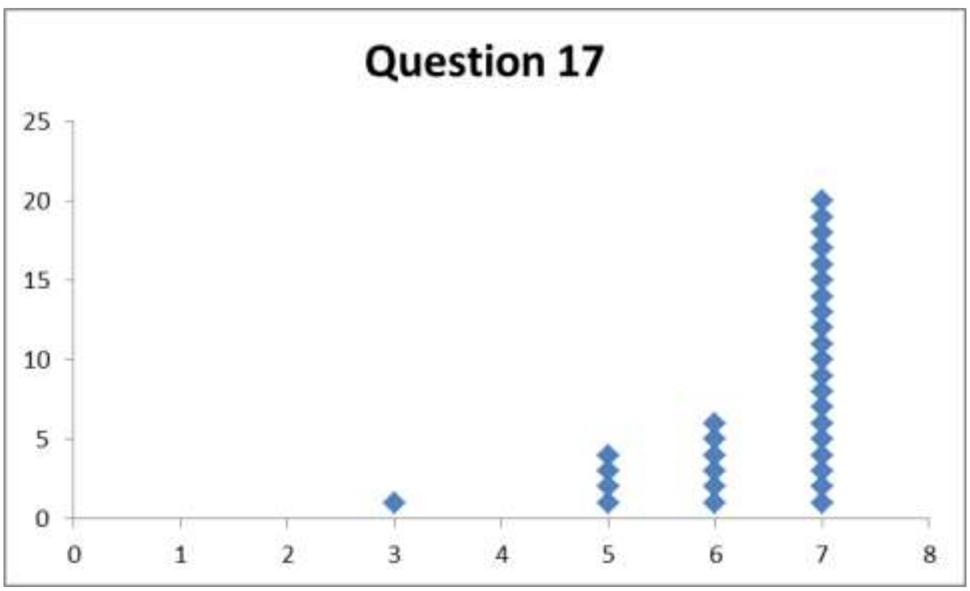




17) Draw the tree below as a rooted tree with root “c”. (7 points)

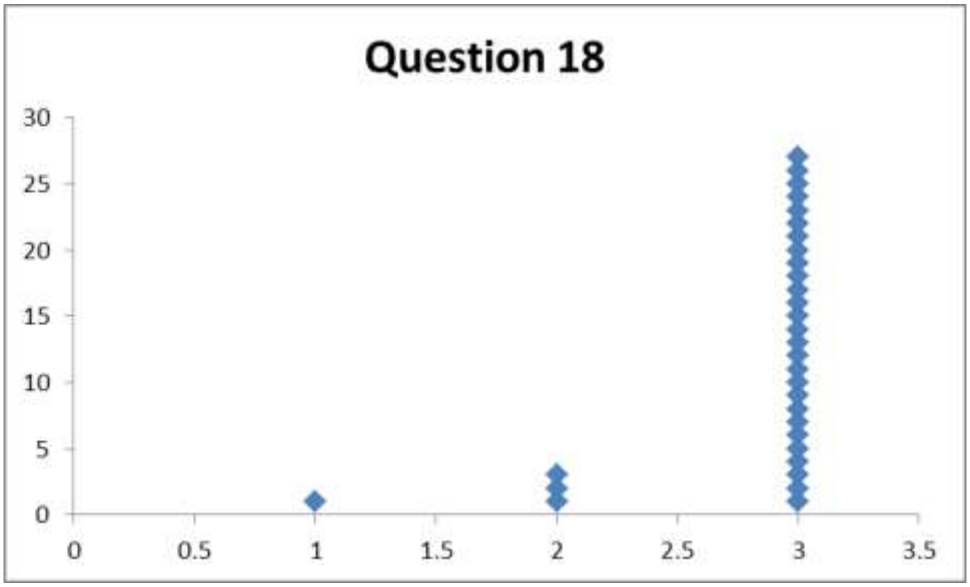






18) In your tree, what are the children of B? (3 points)

K, A, and F



19) In your tree, what is the parent of B? (3 points)

C