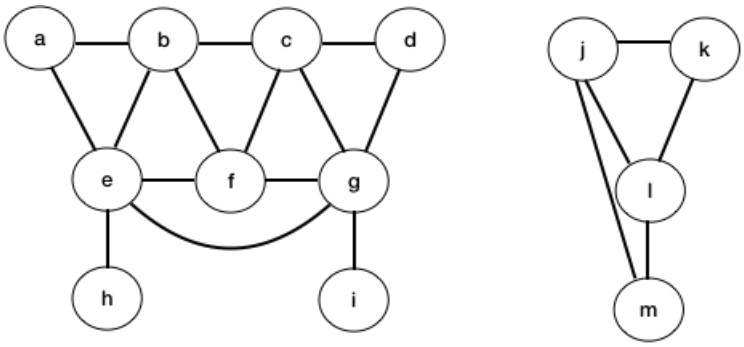
1. In class we used the Blum-Floyd-Rivest-Pratt-Tarjan select algorithm to find the k-th smallest value using $n/5$ blocks of size 5, each. Assume for this problem, we are using $n/3$ blocks, each of size 3.
   (a) Find the number of comparisons for each step of the algorithm.
   (b) Obtain the overall recurrence equation for the upper bound on the number of comparisons and solve it.
   (c) What are your observations about the difference in the asymptotic runtime between a block of size 3 vs. 5 (the one we did in class)? Explain the reason for those differences briefly.

2. In our analysis of the select algorithm for a block size of 5, we compared the *median of the medians* with every other element in the input array to partition it. However, that led to more comparisons than we should have done. We could reduce it since we know that at least $3n/10$ elements are less than or equal to the *median of the medians* and similar number of elements is at least greater than or equal to the *median of the medians*. Obtain a new upper bound on the worst-case number of comparisons using this piece of information. What is the value of the constant?

3. Consider the following graph, $G = (V, E)$, with $|V|$ vertices and $|E|$ edges.



   If we represent graphs in a matrix form, it takes constant time, $O(1)$, to determine whether there is an edge between two vertices, however, the space complexity is $O(|V|^2)$. Representing a graph using an adjacency list has a space complexity of $O(|V|+|E|)$, however, determining an edge, $u, v \in |E|$ can have a runtime complexity of $O(deg(u))$. Answer the following questions:

   1. Show an adjacency list representation of the graph.
   2. Show an adjacency matrix representation of the graph.
   3. Suppose we want to take advantage of space complexity of adjacency lists and still have a faster look up of an edge, give a graph representation so that we can determine whether vertex v is adjacent to vertex u in $O(lg(deg(u)))$. You may describe in English and draw the graph representation.

4. A cynosure among a group of n people is a person who knows nobody but is known to everybody else. The goal of this problem is to identify a cynosure. We will model this relationship with a directed graph that is represented with an adjacency matrix. The vertices in this graph are people. A directed edge means the person represented by the tail of the arrow knows the person represented by the head of the arrow.

    1. Answer the following questions:
        (a) Define cynosure in this graph in terms of its degree?
        (b) Can a group have more than one cynosure?
    2. Write pseudo-code for an efficient algorithm to identify a cynosure or determine that the group has no such person. What is the asymptotic runtime of your algorithm?