Graphs

1. [1 pt] For an undirected graph with n vertices and m edges, the degree of a vertex is defined as the number of edges that have the vertex as an endpoint. Suppose the degree of each vertex of the graph <= 4. Which of the following statement(s) are correct?

    1. m <= 4n
    2. m <= 2n
    3. m <= n
    4. m <= n2
    5. none of the above

2. [1 pt] For an undirected graph with n vertices and m edges, the degree of a vertex is defined as the number of edges that have the vertex as an endpoint. Suppose the degree of each vertex of the graph >= 4. Which of the following statement(s) are correct?

    1. m >= 4n
    2. m >= 2n
    3. m >= n
    4. m >= n2
    5. none of the above

3. [1 pt] You are told that an undirected graph has n vertices and n -1 edges, and is disconnected. Which of the following are correct statement(s)?

    1. The graph may not have a cycle
    2. The graph definitely has a cycle
    3. The graph has exactly two connected components
    4. The graph may have more than two connected components
    5. None of the above

4.Design an algorithm to check whether an undirected graph G = (V, E) has an Euler cycle (find out what Euler cycle means). What is the complexity of your algorithm?

5. Consider a directed graph G = (V, E) that has no cycles. Design an efficient algorithm that outputs the vertices in a sequence {v1, v2... vn} such that for all i < j, there is no edge from vj to vi. (Hint: use depth-first search)

6. Suppose you are given a graph that is “almost disconnected”, i.e., the removal of a single edge will disconnect the graph. Design an algorithm to efficiently find this edge.

7. What are the kinds of graphs for which depth-first search orderings and breadth-first search orderings are the same?

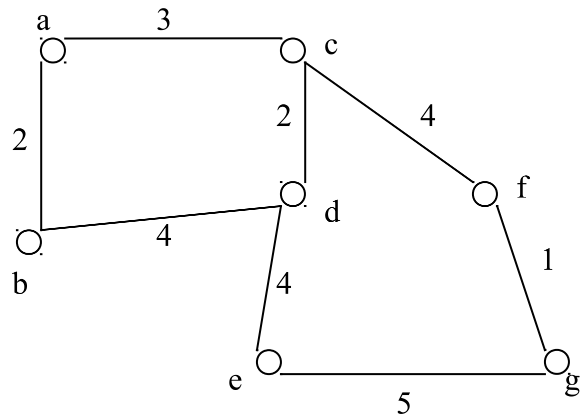
8. [1 pt] For an undirected graph with n vertices and m edges, the degree of a vertex is defined as the number of edges that have the vertex as an endpoint. Suppose the degree of each vertex of the graph is 2. Which of the following statements is correct?

    1. m = 2n

2. m = n
3. m = n/2
4. m = n^2
5. none of the above

Minimum Spanning Trees

1. [2 pt] Run Kruskal's MST algorithm on the following graph



2. [2 pt] Consider an undirected graph with n vertices, and m edges. Assume that the edges are of two types: m1 red edges and m2 green edges. Thus m = m1 + m2. The red edges have weight 1, and the green edges have weight 2. Design and analyze an efficient algorithm to compute the minimum spanning tree of such a graph.

3. Analyze the running time of Kruskal's minimum spanning tree algorithm when the input is an adjacency matrix.

4. [2 pt] A weighted undirected graph with n vertices and m edges is said to satisfy the triangle inequality if for every edge (u, v), the weight of (u, v) is less than or equal to the length of any other alternate path from u to v.

a. Prove that for such a graph, the total weight of all edges is <= (m-n+1)*MST, where MST is the total weight of all edges of the minimum spanning tree.

(Hint: What is the maximum possible weight of an edge of the graph that does not belong to the minimum spanning tree?)

5. A "geometric graph" is a special type of graph where the nodes are points on a 2- dimensional surface and edges are straight lines joining pairs of nodes. Show that the minimum spanning tree of such graphs cannot have edges that cross each other (other than at their endpoints).

6. Suppose you have constructed a minimum spanning tree of a graph (arbitrary graph with arbitrary weights). A new edge is now inserted into the graph. Design an algorithm that will efficiently compute the new MST.

7. [2 pt] Can you simply reverse Kruskal's algorithm to compute the **maximum spanning tree** of a weighted undirected graph? Either give a proof that this works, or show by an example why this will not work.

8. [2 pt] Consider a weighted graph G with n vertices and m edges such that all weights are distinct. Let MST be the minimum spanning tree of this graph.

1. Suppose you insert another edge into this graph, and let MSTInsert be the new minimum spanning tree of this graph. What is the minimum number of common edges between MST and MSTInsert?
2. Given G, MST, and the new edge, design and analyze an efficient algorithm to compute MSTInsert.
3. Instead of inserting, suppose you delete an edge from this graph, and let MSTDelete be the new minimum spanning tree of this graph. What is the minimum number of common edges between MST and MSTDelete?

Shortest Path:

1. Design an example of a graph where the shortest path tree is longer than the minimum spanning tree. In the worst case, how much longer can the shortest path tree be than the minimum spanning tree?
2. A "geometric graph" is a special type of graph where the nodes are points on a 2-dimensional surface and edges are straight lines joining pairs of nodes . Show that the shortest path tree of such graphs cannot have edges that cross each other (other than at their endpoints).
3. Can you modify Dijkstra's shortest path finding algorithm so that it can be used to find longest path in a graph?
4. [2 pt] Consider an undirected graph with n vertices, and m edges. Assume that the edges are of two types: m1 red edges and m2 green edges. Thus m = m1 + m2. The red edges have weight 1, and the green edges have weight 2. Design and analyze an efficient algorithm to compute the shortest path.