MA/CSSE 473 Day 12 Announcements and Summary

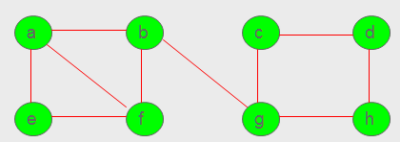
Announcements:

1. **HW6 major changes: Due date postponed until Monday; problems 9-11 have been added back in.**
   - **No late days may be used for this assignment, since the exam is the next day.**
   - **History of this assignment:** When I modified HW6 for Fall, 2014, I removed the last three problems because I didn't think we would get to the material in class soon enough. (I forgot to remove the "It's a substantial assignment" note). Now I see the same issue with a couple of the problems that I left in the assignment. To make sure that you have some "sink-in" time, I postponed the assignment by four days. Now we should have plenty of time to cover the material for the problems that I originally cut out, so I added them back in. The assignment is again substantial, but you have several additional days to get it done.
2. Exam1 date: Tuesday Sept 30,
   - If you are allowed extra time for exams and plan to use that time, please talk with me soon about timing.
3. In my office today: 12:45-5:00.

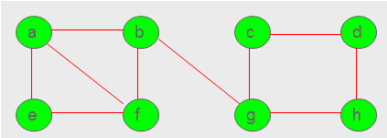
Main ideas from today:

1. Some "decrease by one" algorithms:
   a. Insertion sort, Selection Sort
   b. Depth-first search of a graph, breadth-first search of a graph
   c. Subset generation, permutation generation
2. Breadth-first search and depth-first search are graph traversal algorithms.
   a. Depth-first search (DFS) uses a stack to keep track of unvisited nodes; breadth-first (BFS) uses a queue.
   b. Analogous to pre-order and level-order traversals of a tree.
   c. DFS goes deep, quickly. BFS searches nearby nodes first.
   d. In a connected, undirected graph, both generate a tree and "back edges"`.

3. For the following undirected graph, do a DFS search (starting at a, and always preferring nodes whose names come earlier alphabetically). Show the stack, and the order of pushes and pops.



4. For the same undirected graph, do a BFS search (starting at a, and always preferring nodes whose names come earlier alphabetically). Show the queue, and the order of enqueues and dequeues.



5. In an ***undirected graph***, each edge goes in only one direction. A ***dag*** is a directed acyclic graph, i.e. a directed graph with no (directed) cycles.

6. **Topological sort**: In a dag, the vertices can be linearly ordered so that every edge's starting vertex is listed before its ending vertex.

7. Two Topological sort algorithms:
    a. Based on DFS.
        i. Do a DFS, keeping track of the order in which the nodes are popped off the stack.
        ii. Reverse the order.
    b. Source removal algorithm.
        i. Repeatedly identify and remove a source node. If there are no cycles, there will always be a source.

8. Application of topological sort: Spreadsheet formula evaluation.

9. Generate all permutations of the numbers 1, 2, …, n. Bottom up. Minimal change algorithm.