CSE 241 Algorithms and Data Structures **Spring Semester, 2006**

Homework 5

April 18, 2006 *Due Date: April 27*

1. (5 pts) Complete the on-line course evaluation that is found at `http://evals.wustl.edu/`. I read these carefully and adjust my course according to the comments. So along with check the boxes, please spend a little time answering the questions. It is really helpful for me to both know what portions of the course you really liked as well as your suggestions for improvement. Thanks.

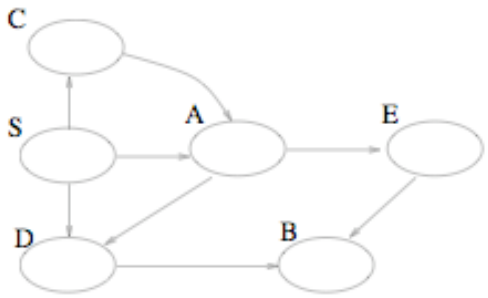
2. (5 pts) Below is the table showing memory where 0 represents `null` and the other memory locations are labeled by addresses 1 to 8. Each memory cell has two fields (left and right). Any value that is a number is a pointer/reference.

cell	left	right
1	7	4
2	0	7
3	4	2
4	b	2
5	3	8
6	0	1
7	a	4
8	7	1

   (a) (3 pts) Draw memory as a graph with a vertex for memory cells 1 to 8. Please have the left pointer originate from the left half of the vertex, and the right pointer originate from the right half of the vertex.

   (b) (2 pts) Suppose that the only variable held by the program is a reference to cell 5. What memory cells are garbage?

3. (15 pts) Consider the following directed graph. You should assume that all lists of adjacent vertices are in alphabetical order.

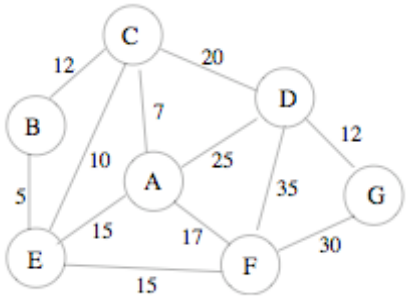


   (a) (6 pts) Show the breadth-first search tree with S as the source.

   (b) (6 pts) Show the discovery and finishing time for all vertices when DFS is run (visiting the vertices in the top-level DFS method in alphabetical order).

   (c) (3 pts) Suppose S, A, B, C, D, and E represent tasks that must be completed as part of a process and that an edge from X to Y means that X must be completed before Y. Name the algorithm you would use to find a valid order to complete the tasks. Then give the valid order that would be output for the graph shown above.

4. (15 pts) In the graph below each vertex represents an island and each edge would be the cost (in millions of dollars) to build a bridge to connect those two islands via a road. You must find a way to build bridges so that one can drive between any two islands with the goal of minimizing the construction cost.



(a) (5 pts) State the name of the algorithm you will use to solve this problem. You may have a choice – pick whichever you want as long as it solves the given problem.

(b) (10 pts) Illustrate the execution of the algorithm you selected on the above graph. Show your work in enough detail that we can see how you reached your solution. (1/4 to 1/2 a page should be adequate – you need not illustrate the details of any data structures used)

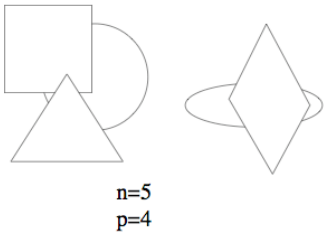
5. (15 pts) Two students have a container filled with 8 liters of a beverage they wish to divide evenly between them. They have two containers with capacities of 3 and 5 liters, respectively. These containers can only be used to measure their capacity (i.e. you may NOT fill the 3 liter container a third of the way to measure 1 liter). However, you can do something like filling the 5 liter container and then use it to fill the 3 liter container. You then know that the 5 liter container holds 2 liters. No other measuring device is available. The students wish to find the minimum sequence of pours that end with two containers each holding exactly 4 liters.

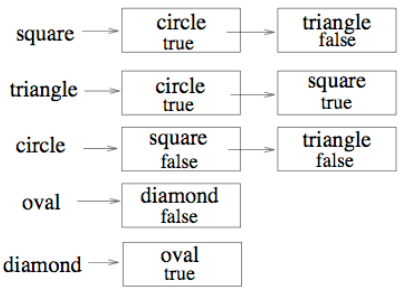
   (a) Represent the set of possible solutions using a graph. Specifically, create a vertex $v = (v_1, v_2, v_3)$ for each triple of non-negative integers that sum to 8 where the first number (v_1) represents the amount in the original container, the second number (v_2) represents the amount in the 3 liter container, and the third number (v_3) represents the amount in the 5 liter container.

   You are to complete the description of the graph by describing that edges you would put in the graph. Be sure to indicate if the edges are directed or undirected, and whether they are weighted or unweighted.

   (b) Name the graph algorithm that would most efficiently determine the fewest pours needed to solve the problem. You should be specific about describing any parameters required by the algorithm you choose and also clearly state how the sequence of pours in the optimal solution is found.

6. (20 pts) Consider the task of redrawing a set of graphic objects in a drawing program. Assume there is an iterator available to iterator over all graphic objects and for each graphic object x there is a list of objects it overlaps with (which can you iterate over) where each list element has two data fields: the object o that x overlaps, a boolean indicating if x is on top of o. Let b be the number of objects and p the number of pairs of objects that overlap. Below is a simple example showing the lists for each of 5 objects in the picture.





When an object is displayed it is drawn in its entirety and will cover anything already drawn that occupies the same space. You are to select/design the most efficient algorithm you can to determine the order in which to display the objects so they appear correctly (or report that it is not possible to do this while drawing an entire object at a time). Then analyze the worst-case asymptotic time complexity of your algorithm (given that the data structure described above is already built) in terms of b and p. If you build any additional data structures then include that in the time complexity analysis for your algorithm.

7. (25 pts) Let G be a directed weighted graph, and let s and t be two vertices in G. Define the bottleneck $b(p)$ of a path p to be the minimum weight edge on the path. That is $b(p) = \min_{e \in p} w(e)$. Describe how you could modify Dijkstra's algorithm to find a path from s to t with the maximum bottleneck. *Hint: Since this is a maximization problem you'll want to use a max-oriented priority queue and thus* decreaseKey *will be replaced by* increaseKey.

   Your solution should contain each of the following components:

   (a) (5 pts) Describe semantically what value will be held as the key for each vertex in the priority queue. For example, in Dijkstra's algorithm the key for vertex v represents the length of the shortest path found so far from the source to vertex v.

   (b) (10 pts) Give pseudo-code to a similar level that was given in class for Dijkstra's algorithm. (Most of the points will be associated with the portions that change.)

   (c) (5 pts) While I do not expect you to prove correctness, you should argue how the parameter given to `increaseKey` maintains the semantics you described in part (a). For example, in Dijkstra's algorithm when you are examining the edge from u to v, the key for u corresponds to the shortest path from s to u. Thus taking this path from s to u followed by the (u, v) edge results in a path from s to v of length that is the sum of the key for u and the weight of the (u, v) edge and that is exactly the parameter given to the `decreaseKey` method. (You only want to update the key if the path found is better, which for Dijkstra means you are using a `decreaseKey` since a shorter path is preferred.)

Challenge Problems:

8. (4 pts) A *total sink* in a directed graph is a vertex that has no out-edges and one in-edge from every other vertex in the graph. Let n be the number of vertices, and m be the number of edges in the graph. Describe an $O(n)$ algorithm to find a total sink (or report that none exists) when the directed graph is given in the adjacency matrix representation.

9. (6 pts) Suppose that all edge weights in a graph are integers in the range from 1 to 10. Your task is to think of a new way to implement the Priority Queue ADT for this special case so that Prim's algorithm runs in time $O(m + n)$ where n is the number of vertices and m the number of edges in the graph. Be sure to clearly describe your data structure for the priority queue and clearly describe how `insert`, `extractMin` and `decreaseKey` will be implemented. Then analyze the worst-case time complexity of these three methods and from that analyze the time complexity of Prim's algorithm.

   How would you modify this priority queue implementation to get an $O(m + n)$ algorithm for Dijkstra's single-source shortest path problem?