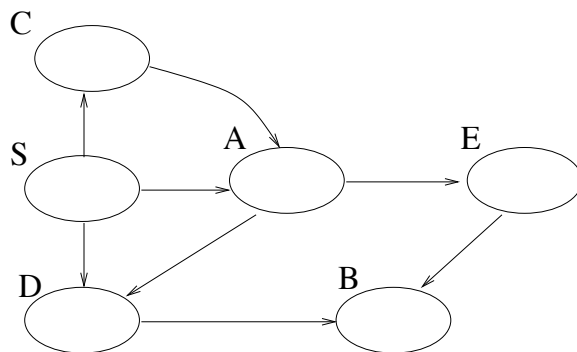


Homework 5

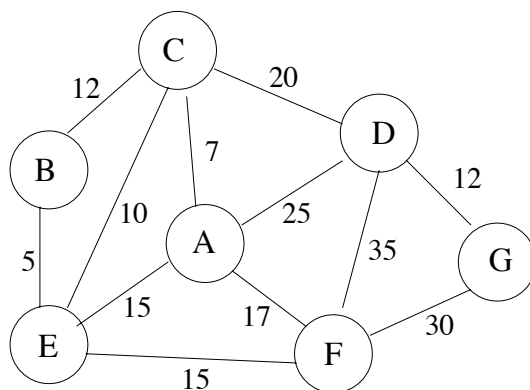
December 6, 2005

Due Date: December 13

1. (15 pts) Consider the following directed graph. You should assume that all lists of adjacent vertices are in alphabetical order.

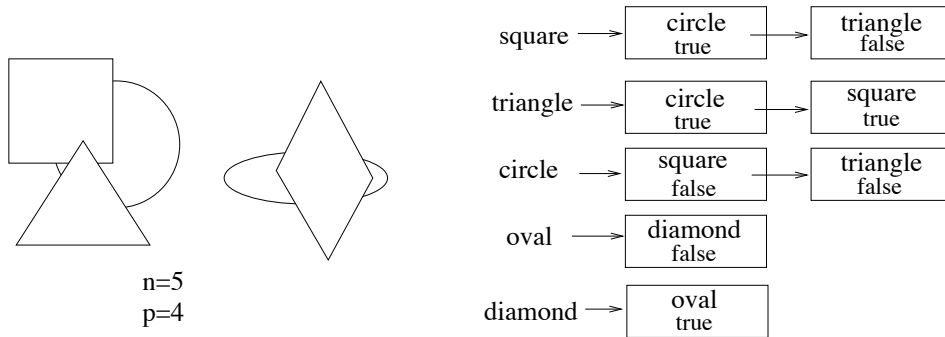


- (a) (6 pts) Show the breadth-first search tree with S as the source.
- (b) (6 pts) Show the discovery and finishing time for all vertices when DFS is run (visiting the vertices in the top-level DFS method in alphabetical order).
- (c) (3 pts) Suppose S, A, B, C, D , and E represent tasks that must 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.
2. (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)
3. (10 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.
- (a) (6 pts) Given a directed graph in the adjacency list representation, given the most efficient algorithm you can to find a total sink (or report that none exists). What is the time complexity of this algorithm?
- (b) (4 pts) Given a directed graph in the adjacency matrix representation, describe in English what the adjacency matrix would look like if there is a total sink.
4. (20 pts) Consider the task of redrawing a set of graphic objects in a drawing program. Assume there is an iterator available to iterate 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.



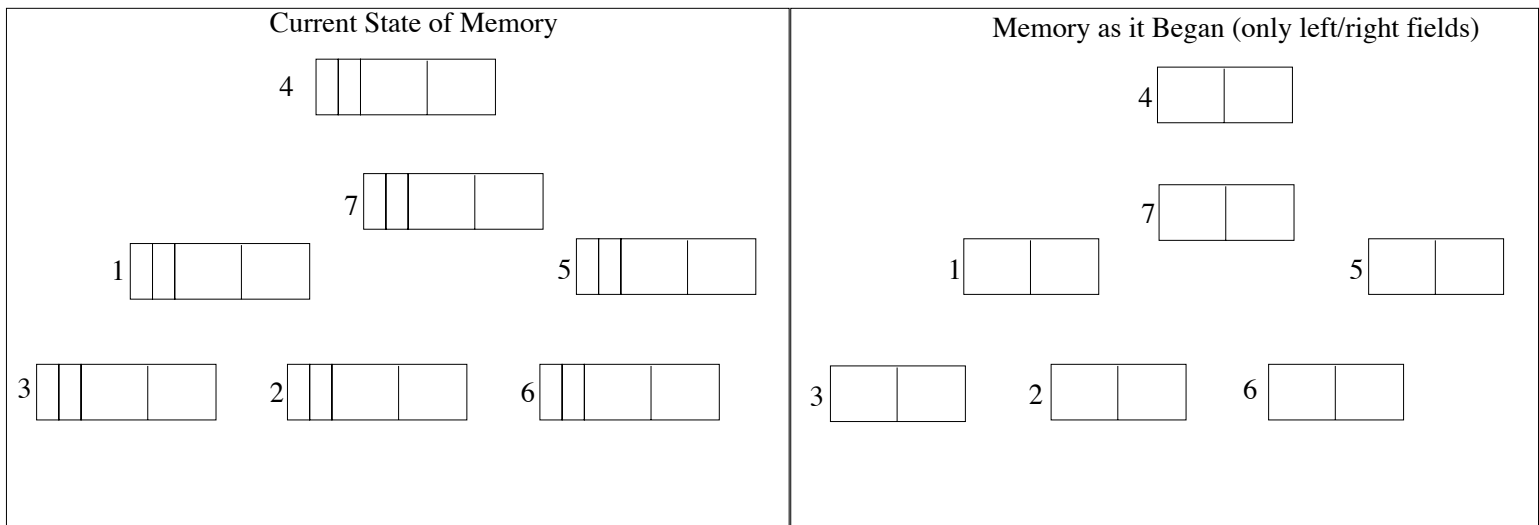
When an object is displayed it is drawn in its entirety and will cover anything already drawn that occupies the same space. You are to select/design the most efficient algorithm you can to determine the order in which to display the objects so they appear correctly (or report that it is not possible 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.

5. (20 pts) A search-engine company has decided on the following method to rank web page quality based on an individual's preference. Each individual can give a web page X that they consider to be a very high quality page. For every other web page its quality is defined as the minimum number of hyperlinks that must be followed from page X to get to it. (In class we will discuss some of the methods really used to rank web page quality). Suppose that there are p web pages being considered with a total of ℓ hyperlinks. You must:

- Formulate this problem as a graph problem. Give enough detail so that given a set of web pages and hyperlinks someone could, without any ambiguities, create the graph. So it should be clear if the graph is directed or undirected, weighted or unweighted, ...
 - Select a graph representation under the assumption that p is roughly 1,000,000 and ℓ is roughly 10,000,000 (i.e. on average each web page as 10 links).
 - Select/Design the most efficient algorithm you can to compute the quality measure defined above for all p web pages and analyze the time complexity of your algorithm.
6. (20 pts) Below is the table showing memory *during* the in-place DFS used by the mark phase of the mark-and-sweep garbage collection algorithm. Note that 0 represents null and “X” is used to indicate the mark bit is set.

Current State of Memory

	mark	back	left	right
1		R	2	7
2	X	L	0	6
3		L	1	d
4		R	a	7
5	X	R	7	6
6	X	R	b	7
7	X	L	2	4



- (5 pts) Complete the graphical view of memory as it is shown in the table on the left.
- (3 pts) Current = 4 and previous = 5. With this information (and memory as shown above) determine the DFS stack.
- (10 pts) Complete the graphical view of memory as it began. Very briefly describe in words below what you did to obtain your solution.
- (2 pts) Which cells will be placed on the free list during the sweep phase? (You can assume that only the one DFS shown in progress is run in the mark phase.)

Challenge Problems:

7. (5 pts) 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.
8. (5 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.

Would you be able to use this same priority queue implementation for Dijkstra's single-source shortest path problem?