

Practice Problems for HW 5 and for the Final Exam

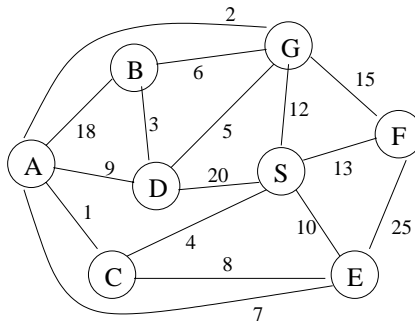
April 15, 2003

- For the following problem, use the directed unweighted graph given by the following adjacency list. Be sure to consider the edges in the given order.

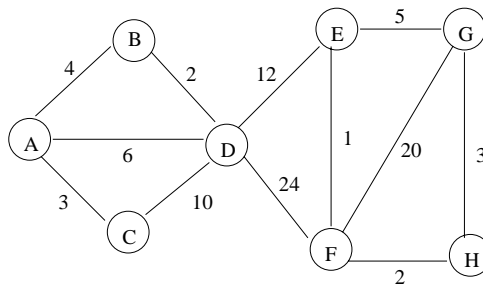
A: C E B  
 B: E D  
 C: E  
 D: C F E  
 E: F  
 F:

- For the source vertex  $s = A$  what is the order in which the vertices are visited by BFS (breadth first search)? Also, show the breadth-first search tree that you obtain.
- What is the order in which the vertices are visited by DFS (depth first search)? You should assume that the top-level DFS procedure visits the vertices in alphabetical order. For each vertex give the discovery and finishing time.
- Suppose that this graph is a precedence graph. Using your work above either give a valid order in which to perform the tasks (call them task A, task B, ..., task F) or prove that there is no valid order.

- Run Dijkstra's shortest path algorithm on the following graph (with  $S$  as the source vertex). Show the resulting shortest path tree and next to each vertex  $v$  in the tree, give the length of the shortest path from  $S$  to it (i.e. " $v.dist$  or in the books notation  $d[v]$ ").



- Consider the following graph:



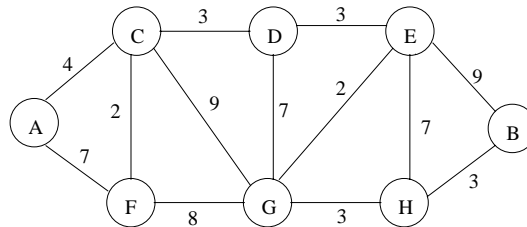
List the edges placed in the minimum spanning tree by Prim's MST algorithm **in the order in which they are added** when the source vertex is  $A$ .

4. Given two specified vertices,  $u$  and  $v$ , in a directed graph  $G = (V, E)$ , you are to compute  $|\{k \in V \mid (u, k) \in E \wedge (k, v) \in E\}|$  (i.e. the number of different intermediate vertices,  $k$ , for which there is an edge from  $u$  to  $k$  and an edge from  $k$  to  $v$ ). Let  $n$  be the number of vertices, and  $m$  the number of edges in the graph.

For both an **adjacency matrix** and **adjacency list** representation of a directed graph, describe how you would efficiently solve this problem. For both representations, clearly describe your algorithm and analyze the worst-case time complexity.

*Note: This operation is used in finding shortest paths when there are negative weight edges.*

5. A region contains a number of towns connected by roads. Each road is labeled by the average number of minutes required for a fire engine to travel it. Each intersection is labeled with a circle. Suppose that you have decided to place your fire station at location  $G$ . (While the example given above is small, you want to automate the process of solving this problem on much larger problems).



- (a) What algorithm would you recommend be used to find the fastest route from the fire station to each of the intersections? Demonstrate how it would work on the example above. Show your work including the resulting routes.
- (b) Suppose one location must be selected for the fire station such that it is chosen to minimize the distance to the farthest intersection. Devise an algorithm to solve this problem given any arbitrary road map. Analyze the time complexity of your algorithm when there are  $f$  possible locations for the fire stations (which must be at one of the intersections) and  $r$  roads.

Note: An algorithm with time complexity  $O(rf + f^2 \log f)$  is sufficient.

6. An airport is developing a computer simulation of air traffic control that maintains events such as landings and takeoffs. Each event has a *time-stamp* that denotes the time when the event occurs.

- (a) Suppose the simulation program needs only to efficiently perform the following two fundamental operations:
- Insert a new event with a given time-stamp (that is, add a future event)
  - Extract the even with the smallest time-stamp (which is the next event to process)

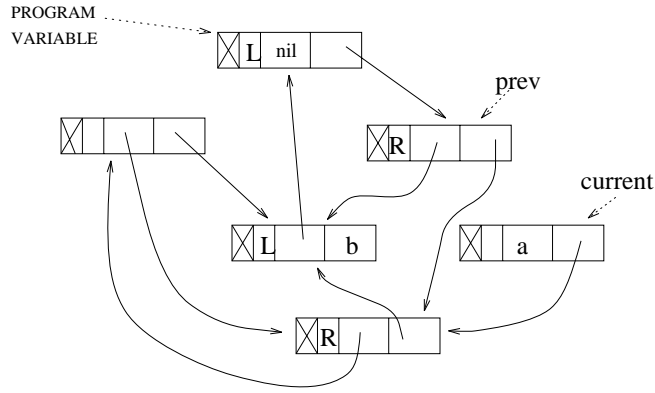
Which data structure would you use for this problem? Justify your answer.

- (b) Suppose that a decision has been made to add the following two operations:
- Given the handle for an event, remove it.
  - Given the handle for an event return the event that precedes it and the event that follows it.

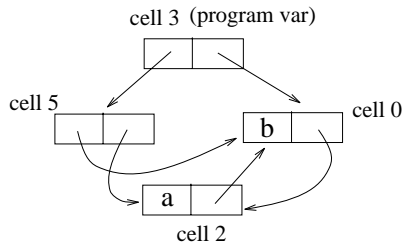
What data structure would you use now? Justify your answer.

- (c) The simulation system (with the operations described in parts (a) and (b)) is now going to be used for a large simulation in which the number of events is so large that most events must be stored on secondary storage. What data structure would you use now? Justify your answer.

7. Below is a pictorial view of memory (only the cells in use are shown) during the middle of the mark phase of the mark and sweep algorithm. Show a pictorial view of memory as it was *BEFORE* the mark and sweep algorithm started executing. (You do not need to show the mark bit and back bit for the cells.)



8. Here we consider copying collection. Suppose that we have 18 memory cells (0, ..., 17). Currently cells (0, ..., 8) are the from-space. Below is a graph showing the configuration of the reachable cells. Note that cell 3 is the program variable.



Below is the current state of memory (with the contents of cells 0,2,3 and 5 missing).

0			9	11	15
1	6	d	10	a	13
2			11	c	c
3			12	9	10
4	a	7	13	16	11
5			14	16	9
6	7	8	15	x	y
7	x	y	16	a	12
8	0	4	17	k	1

- What should be in cells 0,2,3,5?
- Show the contents of memory after phase 1 is completed.
- Show the contents of memory after phase 2 is completed.
- After phase 2, which memory cells are currently available to the program.

9. The below table shows the current state of a memory in which there are 6 cells with addresses 0,1,2,3,4,5. (Each cell has two components where you can assume that an integer value is a pointer and a character value is a data item.)

0	5	3
1	0	3
2	a	3
3	b	2
4	1	c
5	3	2

- (a) Show a pictorial view of memory (in the same style as in Problem 2).
- (b) Given that cell 0 is the only program variable, simulate the in-place DFS used by the mark-and-sweep algorithm showing the pointers, mark bits, and back bits at each step as in the provided handout.

Note: You won't be asked to do a full example like problem like this on the final exam. However, doing part (b) will help you to be sure you understand the in-place DFS algorithm used for the mark phase of both the mark-and-sweep and copying collection garbage collection algorithms.