NAME:

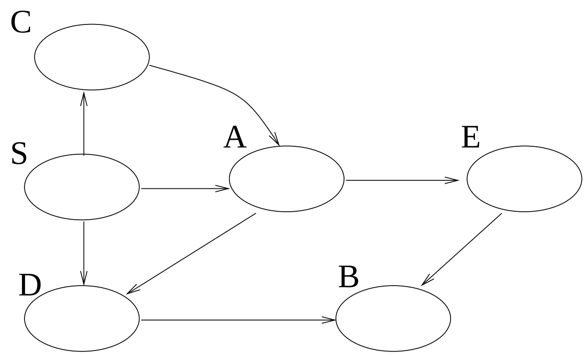
CS 241 Algorithms and Data Structures **Spring Semester, 2004**

Final Exam

May 11, 2004

- Do not spend too much time on any problem. The point value approximates the time I expect you to need for the problem. (Note the exam has 100 points.)
- Please write up all solutions clearly, concisely, and legibly.

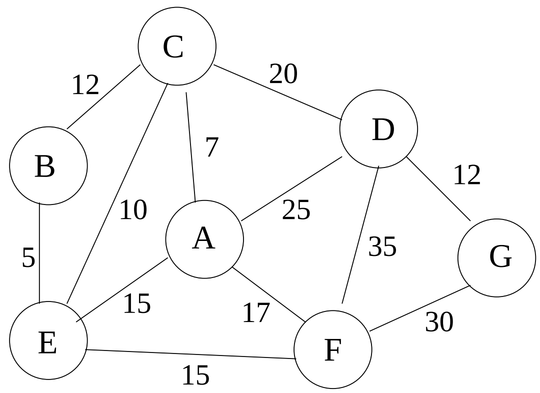
1. (8 pts) Consider the following directed graph. If you iterate through all of the vertices as part of any algorithm then you should go through them in alphabetical order.



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

(b) 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 using the graph above and the space below, show your work in hand simulating this algorithm. Be sure to show your final answer below.

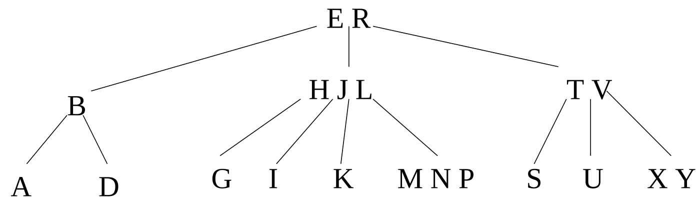
2. (7 pts) In the graph below each vertex represents an island and each edge would be the cost (in millions of dollars) to build a bridge to connect those two islands via a road. You must find a way to build bridges so that one can drive between any two islands with the goal of minimizing the construction cost.



State the name of the algorithm you will use to solve this problem and then demonstrate your solution on the above graph. Show your work to enough detail that we can see how you reached your solution.

3. (15 pts) Let A be a "max-oriented" binary heap. Give the most efficient algorithm you can to compute the number of items with the maximum value. For example, in the binary heap $A = \langle 25, 10, 25, 5, 7, 25, 15, 4, 3, 2, 6, 12, 13 \rangle$, there are 3 items with the maximum value. You should clearly give your algorithm and briefly analyze its time complexity as a function of n, the number of items in A, and k, the number of items with maximum value. You can directly access A.

4. (15 pts) Consider the following B-tree with a minimum branching factor of $t = 2$:



(a) Show the B-tree that results from inserting Q into the above B-tree.

(b) Show the B-tree that results from deleting E in what you obtained AFTER inserting Q.

(c) Draw a legal red-black tree that corresponds to the ORIGINAL B-tree given.

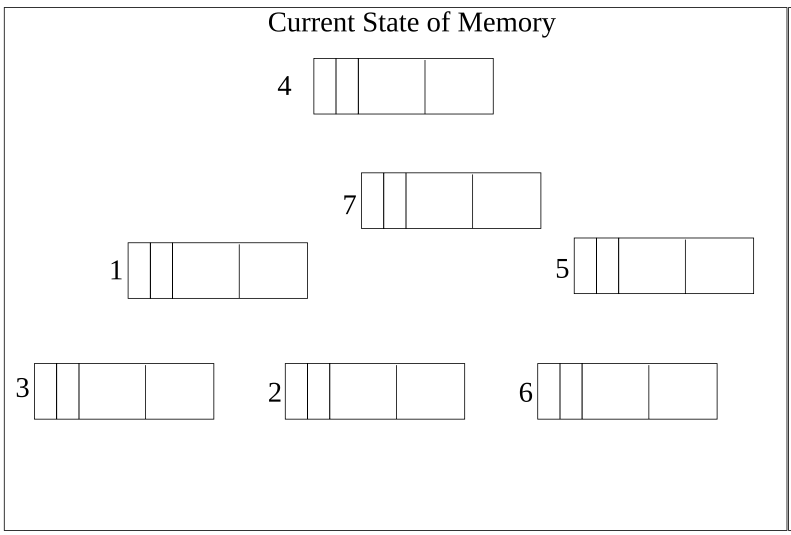
5. (15 pts) Give an $O(n \log k)$ algorithm to merge k sorted lists $L_1, \ldots, L_k$ (sorted in increasing order) into one sorted list, where n is the total number of items in all the lists combined.

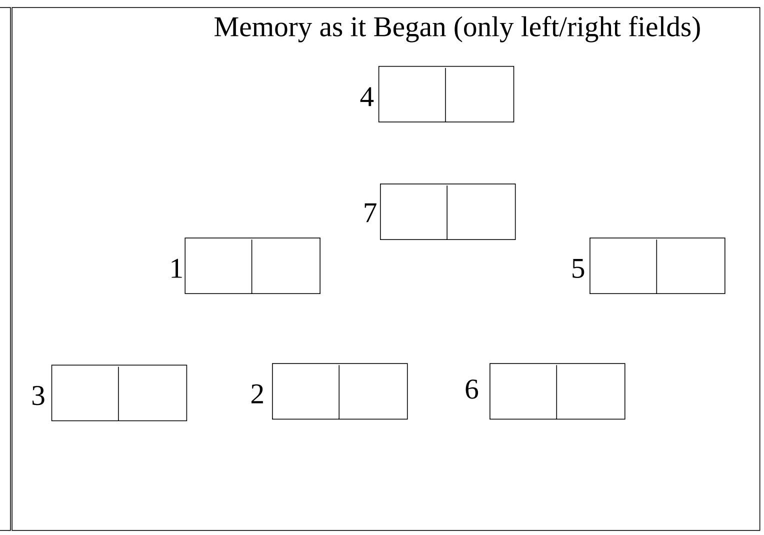
   Big Hint: Think about the merge step of mergesort and how you can use a binary heap. Be sure to clearly describe what you will use as the key and associated data.

6. (10 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





You are to:

(a) Complete the graphical view of memory as it is shown in the table on the left.

(b) Current = 4 and previous = 5. With this information (and memory as shown above) determine the DFS stack and clearly show it below.

(c) Complete the graphical view of memory as it began. Very briefly describe in words below what you did to obtain your solution.

(d) 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.)

7. (20 pts) In this problem you will design data structure(s) for a simple key frame animation system. Remember that you can use any standard methods for data structures we've studied without repeating how they work. For purposes of analysis, let n be the number of frames stored in your data structure. You are to support the following methods.

- `insertFrame(frameID,frame,timeInMovie,artist)` where `frameID` is a unique id for the frame, `frame` is an image, `timeInMovie` is the offset in seconds that the frame will be shown in the movie, and `artist` is the name of the artist who designed the frame. This method should insert the frame and should run in worst-case or expected-case $O(\log n)$ time.
- `produceMovieSegment(startTime,endTime)` produces a list of frames in order of time from `startTime` to `endTime`. This should run in worst-case or expected-case $O(k+\log n)$ time where k is the number of frames in the given time range.
- `getFrames(artist)` should return a list of frameIDs (in any order) for all frames produced by the given artist. To be sure this is clear, you can just return the head pointer to such a list (or an Iterator for such a list). You can assume the artists have unique names. This method must run in worst-case or expected-case $O(1)$ time.

(a) Clearly describe the data structures you would use to implement the above methods so they run as efficiently as possible. While the highest priority is to make these methods run efficiently, also be as efficient in your usage of space as you can. (On the next page you'll describe the methods – here just describe the data structure(s) used.)

(b) Clearly describe the steps taken by `insertFrame` and analyze the time complexity.

(c) Clearly describe the steps taken by `produceVideo` and analyze the time complexity.

(d) Clearly describe the steps taken by `getFrames` and analyze the time complexity.

8. (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) 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) 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. Use the example where there are 5 vertices and the second vertex is a total sink to help illustrate what properties the adjacency matrix will have.

(c) EXTRA CREDIT (5 pts) – ONLY DO THIS IF YOU HAVE TIME Give 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.

Problem	Points Possible	Points Received
1	8	
2	7	
3	15	
4	15	
5	15	
6	10	
7	20	
8	10 (+5)	
total	100	