

## COMPUTER SCIENCE 112 - Fall 2012 FINAL EXAM

Name: \_\_\_\_\_

CIRCLE your section:    W 10:35      W 12:15      W 1:55      Th 3:35      Th 5:15      Th 6:55

---

- Be sure your exam has 6 questions. DO NOT TEAR OFF THE SCRATCH PAGES OR REMOVE THE STAPLE.
- Be sure to fill your name and circle your recitation time above, and your name in all subsequent pages where indicated.
- This is a CLOSED TEXT and CLOSED NOTES exam. You MAY NOT use calculators, cellphones, or any other electronic device during the exam.
- In all questions involving analysis, think carefully and write your answer clearly and precisely, including all the steps that are needed to get the answer. Credit will not be given if your answer comes without a proper sequence of steps with clear explanation.
- NOTE!!! You will only get partial credit for any answer if it is (a) clearly legible and coherent, and (b) close enough to the correct solution that we can infer you know the material reasonably well. A lack of understanding of the concept involved will get little or not credit, even if you have plowed through some mechanics.

**Do not write below this line**

---

Problem -----	Value -----	Score -----
1 Strongly Connected Directed Graph	25	_____
2 Dijkstra's Shortest Paths Algo	25	_____
3 Heap Conversion	25	_____
4 Sorting	25	_____
5 Prefix String Search	25	_____
6 Movie Streaming Website	25	_____
TOTAL	150	_____

### 1. Strongly Connected Directed Graph (25 pts)

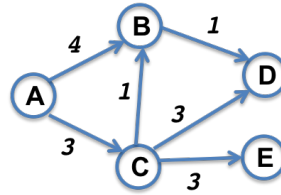
A directed graph is *strongly connected* if every vertex is reachable from every other vertex in the graph. (Note that reachability is via a path that could have one or more edges.) You are given an adjacency linked lists representation of a directed graph. Complete the implementation below to tell if a directed graph is strongly connected. Do not worry about the efficiency of your implementation. You may add other helper methods (with full implementation) as needed.

```
public class DirectedGraph {
    class Edge {
        int vnum;    // vertex number of neighbor
        Edge next;   // pointer to next neighbor
        Edge(int v, Edge ptr) {vnum=v; next=ptr;}
    }
    int n;           // number of vertices
    Edge[] adjLists; // adjacency linked lists
    // define additional fields as necessary

    // returns true if this graph is strongly connected, false otherwise
    public boolean isStronglyConnected() {
        // COMPLETE THIS METHOD
    }
}
```

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

## 2. Dijkstra's Shortest Paths Algorithm (25 pts, 18+7)



a) Dijkstra's shortest paths algorithm is executed on the graph above, starting at A. Assume that vertices A,B,C,D,E are given numbers 0,1,2,3,4 respectively in the implementation. Trace the execution of the algorithm as follows: at the end of every step, show the distance array and the fringe (see table below). The fringe is stored in a min-heap, in which distance updates can be made, apart from delete min, and insert. To show the fringe, draw the binary tree heap structure, with (vertex name,distance) information at each node. Every time there is a change to the heap, show the number of item-to-item comparisons needed to make that change, and also what operation resulted in that change. (Ignore the time needed to *locate* an item in the heap for a distance update.)

Step	Distance array	Fringe heap	Comparisons (number, for what operation)
-----			

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

b) The worst case running time of Dijkstra's algorithm is  $O((n + e) \log n)$ , on a graph with  $n$  vertices and  $e$  edges, provided the fringe is stored in a min-heap with  $O(\log n)$  insert, delete min, and update operations, and the graph is stored in adjacency linked lists. Assume that the graph is stored in an adjacency matrix instead, without changing any other data structure. State clearly which step of the algorithm would be impacted by this change, describe how it would be impacted, and derive the new running time. You don't have to derive the time for the other, non-impacted steps, but you must state all the steps, and write down their big  $O$  running times, then sum to single big  $O$ .

### 3. Heap Conversion (25 pts)

Implement a method to convert a max heap of strings stored in an array into a min heap. Reminder: in a max heap the priority value at any node is  $\geq$  those at its children, while in a min heap, the priority value at any node is  $\leq$  those at its children. The strings are by themselves priorities: if a string comes “before” another in alphabetical order, it is considered to be “less”, and is of lower priority.

The method returns a new array—the original array is NOT modified. Your method must be the *fastest* possible (big  $O$  wise). If it is not, you will get AT MOST HALF the maximum possible credit. You may define and *fully implement* helper methods as necessary. (Fully implemented means you may NOT “call” any method that you have not yourself implemented here.)

```
// Converts a max heap (stored in an array) into a new min heap.  
// Returns a new array that contains the min heap  
// The input array is NOT modified  
public static String[] convertMaxHeapToMin(String[] items) {  
    // IMPLEMENT THIS METHOD
```

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

**4. Sorting (25 pts, 4+5+8+8)**

a) You are given an array with **100,000** entries to sort. Under what circumstances would you use each of the following: (i) insertion sort, (ii) mergesort, and (iii) quicksort?

b) A list of five characters, 'a', 'b', 'c', 'd', and 'e', is stored in an array. Give a starting arrangement of these characters that would result in mergesort making the *most* number of comparisons to sort the list. Show the mergesort recursion tree, as well number of comparisons for each merge, otherwise you will not get any credit. Also, if your answer is not the worst case, you will NOT get ANY credit, even if the process is correctly described. (Assume that a split on an odd number of entries results in one extra entry in the left half.)



198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

Parts (c) and (d) are based on the following:

This year, thousands of students took the SAT (Scholastic Aptitude Test) in New Jersey in preparation to enter college. The total score on the SAT is out of 2400—scores are in multiples of 10. You are asked by the SAT office to write a program that will sort the students in descending values of their scores. If two students have the same score, it does not matter in which order they appear in the sorted result. Assume that the computer on which you will run your program has enough memory to store the names and scores of all the students, either in an array or in a linked list. Also, for this question, you can assume the number of students who took the SAT this year is exactly 65,536, which is  $2^{16}$ .

c) Assuming you were to use mergesort with linked lists, and counting only each score-to-score comparison as a unit time operation, how many such unit time operations (not big  $O$ ) would it take in the worst case to do the sort? Your answer must be simplified to the least possible number of terms.

d) Come up with an algorithm that is better than mergesort above, i.e. it makes fewer unit time operations in the worst case. State what data structure(s) you will use, and clearly describe your algorithm. Then specify what unit time operations you are counting, and determine the total number of such units. You will NOT get any credit if your algorithm is not better than mergesort for this application, even if it works, and you have analyzed it correctly.

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

SCRATCH PAGE

### 5. Prefix String Search (25 pts, 20+5)

a) Implement the following method to return all strings in an alphabetically sorted array that start with a given prefix. For instance, given a prefix “bi”, the returned strings are “Bill Clinton”, “Bill Gates”, and “Bill Joy”. Note that all string comparisons should be case INSENSITIVE. The strings in the returned list must be in the order in which they appear in the array. **Your implementation must be as efficient as possible - if not, you will get AT MOST HALF the credit.** Assume that the array has no duplicate entries. If there are no matches, you may either return null, or an empty array list.

You may use the following `String` methods (in addition to any others you may recall):

```
boolean startsWith(String s)
int compareTo(String s)
int compareToIgnoreCase(String s)
String toLowerCase(String s)
String toUpperCase(String s)
```

(As for `ArrayList`, you only need to use the `add` method to add an item to the end of the array list.)

You may write helper methods (with full implementation) as necessary. You may not call any method that you have not implemented yourself.

```
public static ArrayList<String>
prefixMatch(String[] list, String prefix) {
    // COMPLETE THIS METHOD
```

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

b) Analyze the worst case big  $O$  running time of your implementation: define the parameters that will figure in your big  $O$  result, specify the operations you are counting toward the running time, and show your derivation. (If your algorithm uses another algorithm we have studied in class, you can use the big  $O$  result for the other algorithm without deriving it.)

## 6. Movie Streaming Website (25 pts, 5+8+12)

Credit is for correctness as well as efficiency. If your solution is not the most efficient, you will get AT MOST HALF the credit.

A website streams movies to customers' TVs or other devices. Movies are in one of several genres such as action, drama, mystery, etc. Every movie is in exactly one genre (so that if a movie is an action movie as well as a comedy, it is in a genre called "action-comedy"). The site has around 10 million customers, and around 25,000 movies, but both are growing rapidly.

The site wants to keep track of the most popular movies streamed. You have been hired as the lead engineer to develop a tracking program.

i) Every time a movie is streamed to a customer, its name (e.g. "A Very Harold & Kumar 3D Christmas") and genre ("Comedy") is sent to your program, which then updates the data structures it maintains. (Assume your program can get the current year with a call to an appropriate Java class, in  $O(1)$  time.)

ii) Customers want to know what were the top  $k$  most streamed movies in genre  $g$  in year  $y$ . (If  $y$  is the current year, then accounting is done up to the current date.) For example, what were the top 10 most streamed comedy movies in 2012? Here  $k = 10$ ,  $g = \text{"comedy"}$  and  $y = 2012$ . This query is sent to your program which should output the top  $k$  movie names, in descending order of number of times streamed.

a) Describe the data structures used to implement the above.

b) For (i), describe the algorithm, step by step, to update the data structures. Analyze its big  $O$  running time. Show your work. Specify whether the running time is worst case or something else.

198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

c) For (ii), analyze the big  $O$  running time to output the top  $k$  streamed movies. Describe additional data structures, if any, used in this process. Show your work. Specify whether the running time is worst case or something else.

SCRATCH PAGE



198:112 Fall 2012 Final Exam; Name: \_\_\_\_\_

SCRATCH PAGE

SCRATCH PAGE