

# COMPUTER SCIENCE 112 - Fall 2011

## FINAL EXAM

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

CIRCLE your recitation:    WED 10:35        WED 12:15        TUE 6:55        THU 6:55

---

- Be sure your exam has 7 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.

**Do not write below this line**

---

Problem -----	Value -----	Score -----
1 Quicksort (Simplified Split)	15	_____
2 M-ary Tree	15	_____
3 Topological Sorting using BFS	20	_____
4 Dijkstra's Algorithm	25	_____
5 Hash Table	25	_____
6 Heap Merge	25	_____
7 Movie Streaming Web Site	25	_____
TOTAL	150	_____

### 1. Quicksort (Simplified Split) (15 pts, 10+5)

You are given the following simplified version of split to be used in quicksort:

```
int split(int[ ] A, int lo, int hi) {
    int sp = lo; int pivot = A[lo];
    for (int i=lo+1; i <= hi; i++) {
        if (A[i] < pivot) {
            sp++;
            // swap
            int temp = A[i]; A[i] = A[sp]; A[sp] = temp;
        }
    }
    // swap
    A[lo] = A[sp]; A[sp] = pivot;
    return sp;
}
```

Given the following array of integers:

28    10    8    30    32    37    41    22    45

a) Execute quicksort on this array using the above split method. Show the resulting sort tree, in which each node is the (sub)array to be sorted at that level - the root is the full array. Next to each node write the number of array-item-to-pivot comparisons and data swaps (“// swap” in code) required to split it.

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

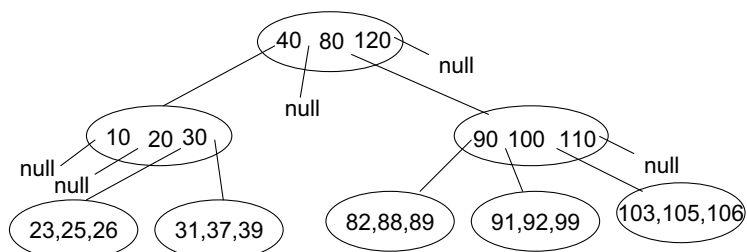
b) In part (a), if insertion sort (and not recursive quicksort) were to be applied on the two sub arrays immediately after the first split, what would be the total number of comparisons to sort? Show work.

## 2. M-ary Tree (15 pts, 5+6+4)

a) An  $m$ -ary tree is a tree in which every node has at most  $m$  children. Consider a special  $m$ -ary tree in which a node either has exactly  $m$  children or no children. What is the minimum number of nodes in a special  $m$ -ary tree of height  $h$ ? (A single node tree has height 0). Show your work, and show how your answer works out correctly on an example special tree with  $h = 3$  and  $m = 3$ .

b) Suppose each node of the  $m$ -ary tree stores  $m - 1$  keys,  $k_1, k_2, \dots, k_{m-1}$  in **ascending order**. Then, all keys  $k$  such that  $k < k_1$  are stored in the first subtree of this node, all keys  $k$  such that  $k_1 < k < k_2$  are stored in the second subtree, and so on. All keys  $k$  such that  $k > k_{m-1}$  are stored in the  $m$ -th subtree.

Here's an example 4-ary tree, with 3 keys in each node:



A search for a target key,  $k_t$ , proceeds as follows (recursive):

0. If root is null, return false. Otherwise, proceed to step 1.
1. Do a sequential search of  $k_t$  against the keys  $k_i, 1 \leq i \leq m - 1$ , at the root, either finding a match, or stopping because  $k_t > k_{m-1}$ , or stopping because  $k_t < k_i$  for some  $i$ . Proceed to step 2.
2. If a match is found, return true. Otherwise, proceed to step 3.
3. If  $k_t > k_{m-1}$ , then recursively search the  $m$ -th subtree and return the result. Otherwise, proceed to step 4.
4. Recursively search the  $i$ -th subtree and return the result.

If the height of an  $m$ -ary tree is  $h$ , how many key-to-key comparisons (exact number, **not** big  $O$ ) would it take in the worst case to successfully search for a target key? Give your answer in terms of  $h$ . Show your work, and show how it would work out correctly for the example 4-ary tree given above.

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

c) Assume a full  $m$ -ary tree, i.e. an  $m$ -ary tree in which every level has the maximum possible number of nodes. If  $n$  is the total number of nodes in a full  $m$ -ary tree, what is the formula for its height,  $h$ , in terms of  $n$ ? Show your derivation.

### 3. Topological Sorting Using BFS (20 pts)

You are given the following algorithm to perform topological sorting on a graph with  $n$  vertices and  $e$  edges using **breadth-first search (BFS)**. Assume the graph is stored in adjacency linked lists format.

1. Let `Pred` be an array of size  $n$ ; let `Q` be an empty queue. Set `topnum` to 0.
2. For each vertex  $v$  in the graph, compute and set `Pred[v]` to the number of edges coming in to  $v$
3. For each vertex  $v$  for which `Pred[v] = 0`, enqueue  $v$  into `Q`. Assign `topnum` to  $v$  and increment `topnum` by 1.
4. While `Q` is not empty, dequeue a vertex, say  $v$ , then decrement `Pred[w]` for each neighbor  $w$  of  $v$ , and finally enqueue  $w$  into `Q` if `Pred[w]` hits 0.

Derive the big  $O$  worst case running time of this algorithm. Show your work by identifying all components that contribute to the running time. If any component takes more than  $O(1)$  time, derive the running time for that component. Then add up the individual running time components, and convert to big  $O$ . Note: You can ignore the time taken to allocate space for objects and arrays. Also, when you derive the running time, you may need to provide more detail than is outlined in the algorithm so your numbers are as specific as possible.

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

#### 4. Dijkstra's Algorithm (25 pts, 18+7)

Here is a weighted directed graph, represented in adjacency linked lists form.

A -> B, 4 -> C, 3

B -> D, 1

C -> D, 3 -> E, 3 -> B, 1

D -> empty

E -> empty

Note: Vertex A is numbered 0, B is 1, C is 2, D is 3, and E is 4.

a) Dijkstra's shortest paths algorithm is executed on this graph, starting at A. Trace the execution of the algorithm as follows: at the end of every step, show the distance array and the fringe. Initially, the distances are all infinity, and the fringe is empty. The fringe is stored in a min heap. Assume that a heap entry can be located in constant time if its distance needs to be updated. To show the fringe/heap, draw the heap as a binary tree, with vertex name and current distance at each node. Every time there is a change to the heap, tell what was done to the heap, show the resultant heap, and show the number of item-to-item comparisons needed to accomplish the change. (If no comparisons were made, write down 0 comparisons.)



198:112 Fall 2011 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 heap with  $O(\log n)$  insert, delete-top, and update operations, and the graph is stored in adjacency linked lists. Suppose 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, how it would be impacted, and show how you arrive at the new running time.

## 5. Hash Table (25 pts, 8+11+6)

You are given the following classes:

```
class LLNode {
    String key; String value; int hashCode; LLNode next;
    LLNode(String k, String v, int h, LLNode nxt) {
        key=k; value=v; hashCode=h; next=nxt;
    }
}

class Hashtable {
    LLNode[] table;
    int numValues;
    float loadFactorThreshold;
    Hashtable(float loadFactorThreshold) {...}
}
```

a) Implement a method in the Hashtable class to insert a key-value pair into the hash table, using the function  $k \bmod N$  to map a hash code  $k$  to a table location.  $N$  is table size (capacity):

```
// inserts (key,value) into hash table,
// calls rehash method (part b) if load factor threshold is exceeded
public void insert(String key, String value) {
```

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

b) Also implement a `rehash` method, which doubles the table size when expanding it. NO new nodes should be created.

```
private void rehash() {
```

c) Suppose you insert 125 integer keys into a hash table with an initial capacity of 25 and a load factor threshold of 2. The hash code is the key itself. Assume the function `key mod table_capacity` is used to map a key to a table position. How many total units of work will be done by the time all keys are inserted if it takes one unit of work to do the mapping, and one unit of work to insert an entry into a linked list? Assume a rehash doubles the table capacity, and that the load factor is checked AFTER an entry is inserted. Ignore the work done to create an array and/or to initialize an array to nulls. Show your work.

## 6. Heap Merge (25 pts)

Suppose you are given two max heaps, stored in arrays.

Write a method to merge them into a single max heap, and return this heap. The original heaps are not modified. Credit is for correctness AND efficiency. You may NOT assume any heap methods - implement everything you need to get your merge to work.

```
// merges max heap1 and heap2 into a new heap that is returned
// heap1 and heap2 are NOT modified
public static <T extends Comparable<T>>
T[] merge(T[] heap1, T[] heap2) {
    // COMPLETE THIS METHOD
```

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

## 7. Movie Streaming Website (25 pts)

(Credit is for correctness as well as efficiency.)

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. "Harold and Kumar: Escape from Guantanamo Bay") and genre ("Comedy") is sent to your program so it can update 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) Also, every once in a while, 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 2010? Here  $k = 10$ ,  $g = \text{"comedy"}$  and  $y = 2010$ . This query is sent to your program which should output the top  $k$  movie names.

Describe the data structures and algorithms used to implement both requirements. For (i), analyze the big  $O$  running time to update the data structures, and for (ii) the big  $O$  running time to output the top  $k$  streamed movies. Show your work. Also specify whether the running time is worst case or average case.

198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

SCRATCH SPACE



198:112 Fall 2011 Final Exam; Name: \_\_\_\_\_

SCRATCH SPACE

SCRATCH SPACE