UNIVERSITY OF WATERLOO
David R. Cheriton School of Computer Science
Midterm CS 240 Fall 2007

Course:	CS 240
Title:	Data Structures and Data Management
Section(s):	2
Instructors:	Therese Biedl and Matthew Nichols

Student Name:

Student ID Number:

Date of Exam:	November 1, 2007
Time Period	Start time: 4:30 pm End time: 6:20 pm
Duration of Exam:	1 hour 50 minutes
Number of Exam Pages:	13 (including cover sheet)
Additional Materials Allowed:	One 8.5" x 11" double-sided help sheet
	No Calculators. No Other Additional Materials

Instructions

- Please sign this exam below these instructions.
- Please ensure that the information recorded on the exam label is correct.
- Cheating is an academic offense. Your signature on this page indicates that you understand and agree to the University's policies regarding cheating on exams.

Your Signature:

Problem	Your Mark	Out of	Marker
1		18	
2		6	
3		4	
4		12	
5		6	
6		4	
7		5	
8		7	
9		5	
Bonus			
Total		67	

Problem 1 (18 **marks)**

For each question below, indicate in the box provided whether the statement is **True** or **False**. No explanation is required. Please do not simply write T or F as these as easily misread. You will receive 2 marks for a correct answer, 0 for a blank answer, and -0.5 for a wrong answer. Your minimum total mark is 0.

a. [ ] If $f(n) \in O(g(n))$ then $f(n) \in o(g(n))$.

b. [ ] Traversing any heap (in binary-tree form) in pre-order will always yield the stored keys in decreasing order.

c. [ ] Depending on the input data, insertion sort may require fewer comparisons than heapsort.

d. [ ] When using external memory, the number of accesses to external memory is far more important for time complexity than the number of operations done in internal memory.

e. [ ] In a randomly built binary search tree, all elements are equally likely to be the root.

f. [ ] A heap of n elements can have height greater than an AVL tree of n elements.

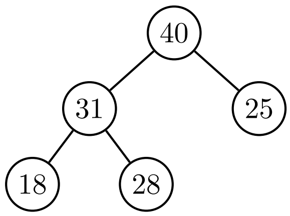
g. [ ] If your professor asks you to develop an algorithm that takes $O(n^2)$ time, then submitting an algorithm that takes $O(n)$ time is acceptable.

h. [ ] For double hashing, the choice of hash functions is not that important since all hash operations can be done in $\theta(1)$ worst-case time regardless.

i. [ ] In extendible hashing, most operations require only 1 or 2 pages accesses from external memory.

Problem 2 (6 **marks)**

a. Consider the max-heap given below. Show the heap that results after inserting the key **10** into this heap. (Note that key **10** is smaller than every other key in the heap).



b. Suppose we have a max-heap H that contains n elements. H is stored as an array A, with the elements of the heap at indices $1..n$. Now we insert into H a key k that is smaller than all other keys in H. At which index of array A will k be located after we have fully completed the insert operation? (Note that array indices start at 1, not at 0). You may assume array A is large enough to accommodate the additional element. Briefly explain your answer.

Problem 3 (4 marks)
Suppose we want to sort the data set 3, 9, 2, 14, 5, 12, 17, using quicksort. In order to have the best-possible running time, which element would have to be the pivot in the initial recursion? Briefly explain your answer.

Problem 4 (12 marks)
For each of the following situations, indicate which sorting algorithm you would choose to implement. You must choose one of the following options: Heapsort, Count Sort, Insertion Sort, Randomized Quicksort, Radix Sort, Selection Sort, Mergesort. Briefly explain each of your answers. Note that the answer may not be unique; your explanation will be the most important part in determining the marks you receive.

a. You have to implement a sorting algorithm in a data base that stores student records. Students are stored using their student ID numbers as keys; these student ID numbers always have 8 digits.

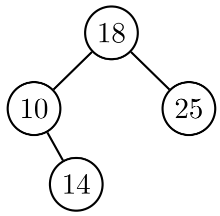
b. You want to sort students by the grade that they have achieved in a CS class. Grades are integers ranging from 0 to 100.

c. Your boss asks you to implement “a sorting algorithm”, but cannot give you any further information as to what type of data you can expect, beyond that you can compare elements.

d. You maintain a data base of employees sorted by their salaries. Initially the data base is already sorted. Now it is time for performance reviews; most salaries will stay the same, but some will change. You need to implement an algorithm to re-sort the data base.

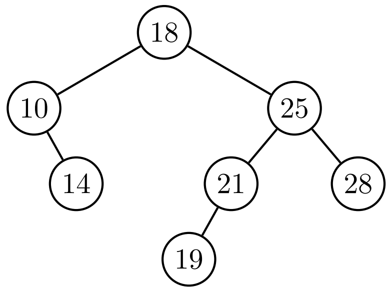
Problem 5 (6 marks)

a. Consider the following AVL-tree:



Insert key **6** and then key **9** into this AVL-tree. Rebalance, if necessary, as described in class. Show the resulting AVL-tree.

b. Consider the following AVL-tree.



Delete the node with key **10** from this AVL-tree. Rebalance, if necessary, as described in class. Show the resulting AVL-tree.

Problem 6 (4 **marks)**

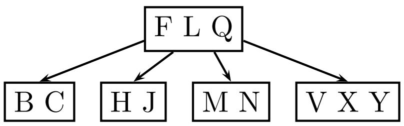
Given the following 15 keys A,B,C,D,E,F,G,H,I,J,K,L,M,N,O.

a. Show a B-tree of order 4 containing these keys that has the smallest possible height. No explanation is needed; only draw the B-tree.

b. Show a B-tree of order 4 containing these keys that has the largest possible height. No explanation is needed; only draw the B-tree.

Problem 7 (5 **marks**)

Consider the following B-tree T of order 4:



a. Insert key **K** into T as described in class. Perform any necessary adjustments. Draw the resulting B-tree.

b. Insert key **W** into T (not into your answer to part (a)!) as described in class. Perform any necessary adjustments. Draw the resulting B-tree.

Problem 8 (7 marks)

Suppose we have a hash table of size 5 and we are given $h(k) = k \bmod 5$ as the hash function.

a. Insert keys 22, 34, 13, and 43 (in this order) into the following hash table using hashing with chaining. Only show the final table after all insertions. The order of elements within each chain is not relevant.

0	
1	
2	
3	
4	

b. Insert keys 83, 52, 44, and 74 (in this order) into the following hash table using linear probing. Only show the final table after all insertions.

0	
1	
2	
3	
4	

c. Insert keys 22, 34, 13, and 43 (in this order) into the following hash table using double hashing, with $h_1(k) = k \bmod 5$ as the hash function, and $h_2(k) = (\text{sum of the digits of } k) \bmod 4 + 1$ as the secondary hash function. Only show the final table after all insertions.

0	
1	

2 $\boxed{}$

3 $\boxed{}$

4 $\boxed{}$

Problem 9 (5 marks)
Examine the following pseudocode and express its runtime using θ-notation. Briefly justify your answer.

```
mystery(int A[1..n], int n) {
   // pre: n is a power of 2
   for i=1..n {
      A[i] = A[i] + 1;
   }
   if (n>1) mystery(A, n/2)
}
```

Problem 10 (Bonus)

Prof. Dumb claims to have developed an implementation of the Priority Queue ADT that guarantees an $O(1)$ running time for both the ExtractMin and Insert operations.

Prove that such an implementation is **not** possible in general (i.e., if you have no extra information about the priorities stored in the priority queue.)