University of Waterloo
Midterm Examination

Fall 2009

Student Name ____________________

Student ID Number ____________________

Course Abbreviation & Number	CS 466/666
Course Title	Advanced Algorithms
Section(s)	01
Held With Course(s)	none
Section(s) of Held With Course(s)	none
Instructor	T. Biedl

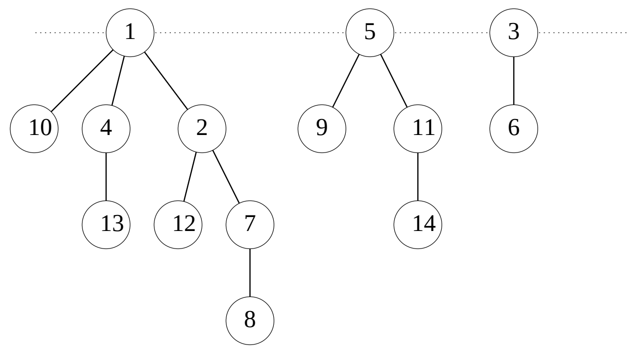
Date of Exam	October 27, 2009
Time Period: afternoon	Start Time: 1:00pm End Time: 2:20pm
Duration of Exam:	1 hour 20 minutes
Number of Exam Pages	8 pages (including this title page)
Exam Type	Special Material
Additional Materials Allowed	1 letter-sized sheet of paper with anything written or typed on both sides.

1. Complete all answers in the spaces provided. Extra space is on the last page.
2. Proctors will only confirm or deny the existence of errors on the exam.
3. All answers must be justified (unless specifically said so otherwise.)
4. If the question is not clear, state any assumption you make.
5. Cheating is an academic offence. Your signature on this exam indicates that you understand and agree to the university's policies regarding cheating on exams.

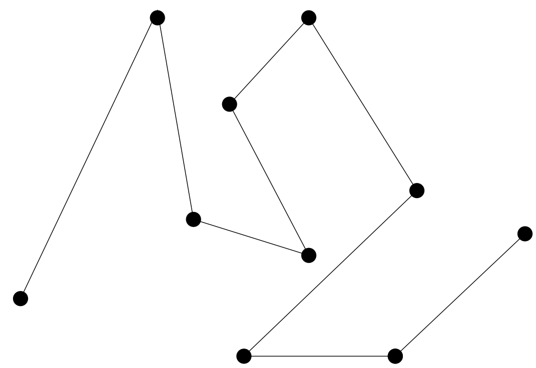
#	Marks	Actual	Initial
1	10		
2	10		
3	10		
4	15		
5	10		
6	15		
7	10		
$\sum$	80		

Signature: ____________________

1. (10 marks): **Binomial heaps:**
The following picture shows a binomial heap. Show the heap that would result from applying *ExtractMin*. (There is more than one possible answer, depending on how you break ties. Only show one answer, i.e., break ties arbitrarily.)



2. (10 marks): **Upper Hull**
   Assume you are given a polygonal chain C, i.e., a sequence of points $p_1, p_2, \ldots, p_n$ such that none of the line segments (p_i, p_{i+1}) intersect (except at common endpoints.) The figure below illustrates a polygonal chain with 10 points.

   Given an algorithm that computes the upper hull of the points of a polygonal chain in $O(n)$ time. You may assume that p_1 has the minimal x-ccoordinate and p_n has the maximal x-coordinate among $p_1, \ldots, p_n$.

3. (10 marks): **Integer Program**

   You need to transport 400 students to Toronto, by car, bus or train. 25 of the students have a car. If a student drives a car you must pay him/her \$20 for gas, but he/she can take 4 other students along (a total of 5 students per car.) You can also rent buses: this costs \$400 per bus, and a bus holds 50 students. Finally you can send students by train: this costs \$20 per student on a train.

   What is the cheapest way to get the students to Toronto, subject to the above conditions? Formulate this problem as an integer program. Clearly state the intended meaning of your variables.

   This question is about the formulation; you need not compute the optimal solution and doing so will not give extra credit.

4. (15 marks): **Sorting via median-finding**

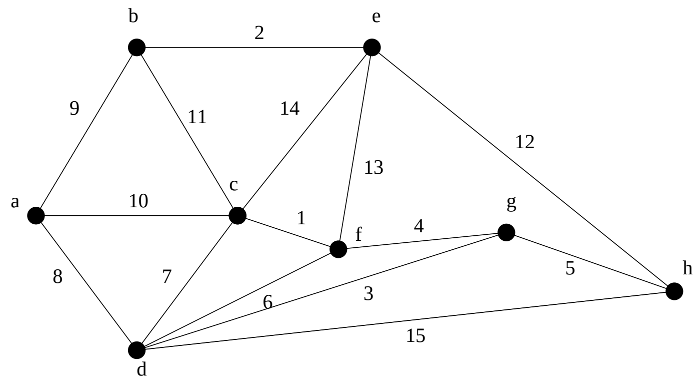
Assume that you have a data structure S that supports the following operations :

- S.Insert(k): Inserts an element with value k into S.
- S.DecreaseKey(x, k): Decreases the key of element x to be k. Nothing is done if k is greater than the current key of x.
- S.ExtractMedian(): Removes and returns the $\lfloor n/2 \rfloor$th smallest element in S, where n is the number of elements currently in S.

(a) Give an algorithm to sort n numbers $a_1, \ldots, a_n$ that uses one such data structure S and $O(n)$ of the above three operations. All other steps of your algorithm should take $O(n)$ time. You may assume that the numbers are positive and distinct.

(b) Give a lower bound on the running time required for a total of n operations Insert, DecreaseKey, and ExtractMedian. (Hint: use (a) even if you did not answer it.)

5. (10 marks): **Borůvka's algorithm**
The following picture shows graph G with edge-weights. Show the graph G' that results after applying one Borůvka-step (for computing the minimum spanning tree.) Process vertices in alphabetical order.



6. (15 marks): **Knights and Knaves**

A far-away island has n people. 90% of these people are knights: they always answer the truth. 10% of these peoples are knaves: they may or may not answer the truth.

(a) Give a (very fast and very simple) Monte-Carlo algorithm to find a knight. What is the error-probability?

(b) Give a (also very simple) deterministic algorithm that tests whether a given person X is a knight. Your algorithm should take $O(n)$ time. (Remark and hint: It takes $O(1)$ time to ask one question to one person on the island.)

(c) Give a Las Vegas algorithm to find a knight. Your algorithm should have expected run-time $O(n)$. (Hint: use (a) and (b) even if you did not answer them.)

7. (10 marks): **True or False?**

For each of the following statements, indicate whether they are True or False. You need not justify your answer.

If you are not sure of the answer, leave it blank. A wrong answer will give **less** credit than no answer.

		TRUE	FALSE
(a)	With Fibonacci-heaps, we can support Insert and Extract-Min in $O(1)$ amortized time.		
(b)	Any comparison-based algorithm that can answer the question "Given a set of n points, does the upper hull have at most 666 points?" must use $\Omega(n \log n)$ comparisons.		
(c)	If Union/Find is implemented with trees and weighted-union heuristic (but not path compression), then the trees have height $O(\log n)$.		
(d)	Let $T(n,m)$ be a recursive function (for any $1 \leq n < m$) defined as follow: $T(n,m) = \begin{cases} 0 \text{ if } m \leq 2n \\ \min_{n \leq d < m}\{T(n,d) + T(d,m) + 1\} \text{ otherwise} \end{cases}$ Then $T(n,m) \in O(\log(m/n))$.		
(e)	The following is a Las Vegas algorithm to find the minimum element among numbers $a_1, \ldots, a_n$: Randomly choose an index $i \in \{1, \ldots, n\}$, and return a_i.		

Extra space below