CS 466/666, Spring 2007
Advanced Algorithms
Timothy Chan

University of Waterloo
Final Examination

Last Name: ________________ First Name: ________________

Signature: ________________________________

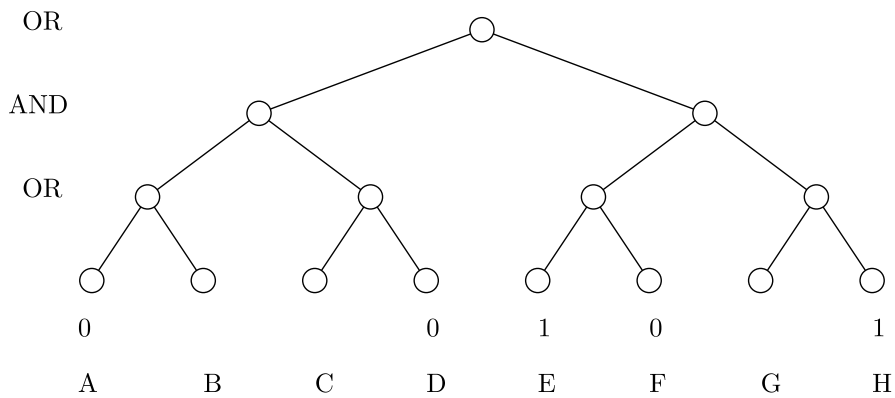
ID number: ________________________________

- Date: August 15, 2007.
- Start Time: 9:00am. End Time: 11:30am.
- Number of pages (including cover and two blank pages): 15.
- No additional materials are allowed.
- Print your initials at the top of each page (in case a page gets detached).
- All answers should be placed in the spaces given. Backs of pages will not be marked and may be used as scratch papers. If you need more space to complete an answer, continue on the blank pages at the end.
- Cheating is an academic offense. Your signature on this exam indicates that you understand and agree to the University's policies regarding cheating on exams.

Q	Marks	Init.
1	/28	
2	/20	
3	/10	
4	/13	
5	/13	
6	/16	
Total	/100	

1. [*28 marks*] *Routine questions.*

   (a) [*8 marks*] Recall the adversary argument from class, showing that to evaluate an AND-OR tree, any deterministic algorithm must examine all leaf values. Consider the following scenario, where the algorithm has already seen the values of the leaves A, D, E, F, and H as shown below:



Suppose the algorithm next examines the leaf B. If you are the adversary, what value would you assign B? Briefly explain.

Continuing on, if the algorithm next examines the leaf G, what value would you assign G? Briefly explain.

(b) [*8 marks*] Write out the linear programming relaxation of the MAX-1-2-SAT problem on the following four clauses:

$$x_1 \vee \overline{x_2}, \quad \overline{x_1} \vee \overline{x_3}, \quad x_2, \quad \overline{x_1} \vee \overline{x_2}, \quad x_1 \vee x_3.$$

Prove that z_{LP}, the value of the optimal real solution, is different from z^*, the value of the optimal integer solution, in this example.

(c) [*12 marks*] Run the four bin-packing algorithms, First-Fit, Best-Fit, First-Fit-Decreasing, and Best-Fit-Decreasing algorithm on the following input:

$$0.3,\ 0.4,\ 0.8,\ 0.05,\ 0.2,\ 0.25$$

First-Fit:

Best-Fit:

First-Fit-Decreasing:

Best-Fit-Decreasing:

2. [*20 marks*] *Short questions.*

   (a) [*4 marks*] True or False: By a data structure from class, we can maintain a set of n numbers so that decrease-key and delete-min both takes $O(1)$ amortized time and insert takes $O(\log n)$ amortized time. Explain.

   (b) [*3 marks*] True or False: $\alpha(n) = o((\log \log n)^{1/466})$. Explain.

   (c) [*3 marks*] True or False: the Seidel–Welzl algorithm for the smallest enclosing circle problem from class is a Monte Carlo algorithm. Explain.

(d) [*4 marks*] Point out two differences between the two random-walk algorithms by Papadmitriou and by Schöning for the satisfiability problem from class.

(e) [*3 marks*] True or False: The analysis of Christofides' factor-3/2 algorithm for traveling salesman problem only applies to the metric case where the triangle inequality holds. Explain.

(f) [*3 marks*] Recall the randomized online paging algorithm RAND-MARK from class. True or False: The $O(\log k)$ upper bound on the expected competitive ratio still holds even if the request sequence is generated by an adversary that knows the random choices made by the RAND-MARK algorithm in advance. Explain.

3. [*10 marks*] *Lower bound by reduction.* Consider the following data structure operations to maintain a set S of n numbers:

   - insert(x): insert a new element x to S;
   - decrease-key(x, k): decrease an element x's key to a given number k;
   - median(): return the $\lfloor |S|/2 \rfloor$-th element in S.

   Show that no data structure can achieve $O(1)$ amortized time for insert(), decrease-key(), and median() simultaneously, under a comparison-based model.

   [Hint: use a reduction involving sorting. You might want to use multiple extra keys like ∞'s or $-\infty$'s.]

4. [*13 marks*] *Randomized approximation algorithms.* We are given an undirected graph $G = (V, E)$ where every vertex has degree exactly 4. In the BALANCED-ORDERING problem, we want to find an ordering L of the vertices, so as to maximize number of *well-balanced* vertices, i.e., the number of vertices v such that exactly 2 neighbors u of v appear before v in L and exactly 2 neighbors w of v appear after w in L.

   (a) [*3 marks*] Fix k distinct elements a, b, c, d, e. If X is a random ordering of a, b, c, d, e (each permutation being equally likely), what is the probability that a is the *middle* element (i.e., has the 3rd position) in the list X?

   (b) [*3 marks*] Design a (very simple!) randomized polynomial-time approximation algorithm for the BALANCED-ORDERING problem.

(c) [*7 marks*] Carefully show that your algorithm has expected approximation factor bounded by a constant (assuming every vertex has degree 4). Specify the constant.

5. [*13 marks*] *On-line algorithms.* Recall the list update problem from class, where we are given an on-line sequence of access requests for a list of k elements, accessing the i-th element of the list costs i units, and swapping two adjacent elements costs c units for a constant $c \geq 1$.

   Consider the following on-line algorithm "Move-Halfway-To-Front" (MHTF):

   when we access element x:
   let i be the position of x (i.e., x is the i-th element in the list)
   move x so that it becomes the $\lceil i/2 \rceil$-th element of the list

   (a) [*5 marks*] Let L denote the current list maintained by MHTF and L^* denote the current list maintained by the optimal offline algorithm OPT.

   Consider one access operation for element x. Let ℓ denote the number of elements at position $\lceil i/2 \rceil, \ldots, i-1$ in L that are before x in L^*. Let m denote the number of elements at position $\lceil i/2 \rceil, \ldots, i-1$ in L that are after x in L^*. Let s denote the number of swaps performed by OPT on this operation.

   Give an upper bound on the cost of MHTF and a lower bound on the cost of OPT in terms of the parameters ℓ, m, and s for this single operation.

(b) [*3 marks*] Define the potential Φ to be the number of inversions between the two lists L and L^*. Give an upper bound on the change in potential in terms of the parameters ℓ, m, and s for a single operation.

(c) [*5 marks*] Now, prove that MHTF has competitive ratio bounded by a constant. Express the constant in terms of c.

6. [*16 marks*] *Approximation algorithms.* We are given a sequence S of n jobs, where the i-th job requires duration $s_i \geq 0$. We want to assign jobs to 3 processors to minimize the completion time. More precisely, we want to partition S into three subsets X, Y, Z so that the cost function $c(X, Y, Z) = \max\{\sum_{s_i \in X} s_i,\ \sum_{s_i \in Y} s_i,\ \sum_{s_i \in Z} s_i\}$ is minimized.

   (a) [*3 marks*] Let OPT be the minimum cost. Let $\sigma = \sum_{i=1}^{n} s_i$ be the total duration. Show that OPT $\geq \sigma/3$.

   (b) [*3 marks*] Let $\varepsilon > 0$ be a constant. Show that there are at most a constant number of elements with $s_i > \varepsilon\sigma$.

(c) [*5 marks*] For the general problem, consider the following approximation algorithm:

1. solve problem optimally by brute force for the jobs with $s_i > \varepsilon\sigma$ and put these jobs in subsets X, Y, Z
2. for each remaining job $s_i \le \varepsilon\sigma$ do {
3. w.l.o.g., assume $\sum_{s_j \in X} s_j$ is smaller than $\sum_{s_j \in Y} s_j$ and $\sum_{s_j \in Z} s_j$ (because otherwise we can swap X, Y, Z)
4. put s_i in X

   }

Prove that $c(X, Y, Z) \le \text{OPT}$ or $c(X, Y, Z) \le \sigma/3 + \varepsilon\sigma$.

(d) [*5 marks*] Using parts (a) and (c), bound the approximation factor of this algorithm. Using part (b), bound the running time of this algorithm. Is this a polynomial-time approximation scheme (PTAS)?

(Extra space)

(Extra space)