Midterm Examination
CS 466 — Advanced Algorithms

October 30, 2002

Naomi Nishimura **Fall term 2002**

First Name:	**Last Name:**
Initials:	**ID number:**
Signature:	

Problem	1	2	3	4	Total
Marks	16	14	10	10	50
Marks					
Marker					

Instructions — Read Carefully

1. Before beginning the examination paper, make certain you have **9** pages, including this page. The last page is blank and can be used if you need additional space for some problem.
2. Place your name, student ID number, and signature in the box on top of the title page.
3. Print your initials in the space provided at the top of each page (in case your exam booklet comes unstapled).
4. You are to answer all problems. The marks assigned to each problem are shown on this page and at the beginning of each problem; use these to organize your time effectively. Please note that the number of marks for a question does not necessarily indicate the difficulty of that question.
5. Read through the entire paper carefully before you start to write, and do the easiest problems first. All solutions must be placed in this exam booklet in the spaces given. Be clear and concise. Indicate briefly how you obtained your answer. You may use the backs of the pages of the exam as scratch paper. Backs of pages will not be marked. If you need more space to complete an answer, please continue on a blank page at the back (and indicate that you have done so, to have it marked).
6. 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.

Problem 1 *[16 marks]*

Consider the up-tree created by a sequence of operations, using path compression and a modification on weighted union in which if two roots have equal rank, the root with the smaller value is made into the child of the root with the larger value. Observe that the tree is unordered, that is, the left to right order of children is arbitrary.

(a) *[8 marks]* Draw the structure created by the sequence of operations `Union(1,2)`, `Union(3,4)`, `Union(2,4)`, `Find(1)`, assuming that all necessary `Make-Set`'s have already been executed. Demonstrate the progression by illustrating the forest resulting from each step in the sequence; you may omit single-node trees to save time and space.

Result after `Union(1,2)`:

Result after `Union(3,4)`:

Result after `Union(2,4)`:

Result after `Find(1)`:

Problem 1 (*continued*)

(b) *[8 marks]* Is there another sequence of operations that yields the structure given in part (a) (assuming again that all `Make-Set`'s have been performed)? If so, give an example of such a sequence, showing the result of each step. If not, answer why there can be no such sequence.

Problem 2 *[14 marks]*

In this question we will consider different methods for obtaining lower bounds for two problems. Some of the methods may result in very weak lower bounds. Consider the problem Majority defined formally below. For any element x in a list, define equals(x) to be equal to the number of elements in the list equal to x. An element x is a *majority element* if equals$(x) > n/2$, where n is the number of elements in the list.

Input: A list X of n not necessarily distinct numbers $x_1, \ldots, x_n$.

Solution: An integer j, $0 \leq j \leq n$, such that x_j is a majority element if $1 \leq j \leq n$ (0 means there is no majority element).

We define the following related decision problem, which we call MajDec.

Input: A list X of n not necessarily distinct numbers $x_1, \ldots, x_n$ and an integer $1 \leq j \leq n$.

Solution: Is x_j a majority element?

(a) *[3 marks]* For each problem, give an information theory lower bound on any comparison-based algorithm that solves the problem. A comparison between two elements returns $=$ or $\neq$. Is one lower bound better than the other? Why? Briefly justify your answers.

Problem 2 (*continued*)

(b) *[6 marks]* Give **simple** polynomial-time reductions from Majority to MajDec and MajDec to Majority. Be sure to specify which is which. Do not spend too much time on this question!

Problem 2 (*continued*)

(c) *[5 marks]* Suppose we had lower bounds ℓ_m on Majority and ℓ_d on MajDec. Consider the reduction from Majority to MajDec that you gave in part (b). Can you use your reduction to obtain a new lower bound on Majority? Can you use your reduction to obtain a new lower bound on MajDec? Briefly justify your answers. *Note: you do not need to change your answer to (b). Your answer to (c) will receive full marks if it makes sense with your answer to part (b).*

Problem 3 *[10 marks]*

Consider the Monte Carlo string matching algorithm given in class.

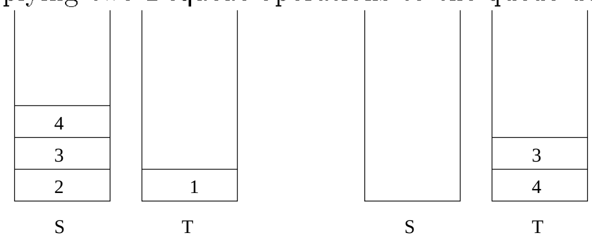
(a) *[4 marks]* The string matching algorithm uses the technique of fingerprinting. Using at most three sentences, describe fingerprinting, listing the advantages and disadvantages of using fingerprinting for string matching.

(b) *[6 marks]* For input pattern P=101, input text T = 00000000000, and primes chosen from the set $\{2, 3, 5, 7\}$, what is the probability of a false match? Briefly justify your answer.

Problem 4 *[10 marks]*

Consider the implementation of a queue with two stacks, stack S and stack T, where the front of the queue corresponds to the top of T, if T is non-empty, and the bottom of S otherwise, and the back of the queue corresponds to the top of S. The operation Enqueue consists of pushing the new item onto stack S, and the operation Dequeue consists of checking if T is empty, if not popping the item on the top of stack T, and if so, first popping each element of S and moving it to T, and then popping the item on the top of stack T.

Note: these are normal stacks with operations `Push-`S, `Push-`T, `Pop-`S, `Pop-`T, `Stacktop-`S, and `Stacktop-`T; each stack operation has a cost of 1 dollar. In the figure below, the queue on the right is obtained by applying two Dequeue operations to the queue depicted on the left.



Use the potential method to show that the amortized cost of each queue operation (that is, both Enqueue and Dequeue) is in $O(1)$.