The University of British Columbia

Computer Science 221: Discrete Math, Data Structures

Midterm Examination, March 2, 2006. Instructor: E.M. Knorr

Time: 75 minutes.

Closed book. No notes. No cell phones. No calculators (you can leave answers in combinatorial form).

Name_____ Student No_____

(PRINT) (Last) (First)

Signature_____

The examination has 7 pages, and that includes this cover sheet. Check that you have a complete paper.

Print your name and ID at the top of this page, and provide your signature. Have your student ID ready.

Since there are serial numbers on these pages, and to save you time, you *do not* have to print your name on each page.

This is a closed book exam. No help sheet, and no calculators are permitted. You can leave your answers in combinatorial form.

Work quickly and do the easy questions first. Part marks are available. **Be sure to show your work.**

The marks for each question are given in braces**.** Do not spend too much time on any one question.

To minimize disruptions during the exam, **please avoid asking the invigilators for help, tips, or explanations.** *Please write down any reasonable assumptions that you are making, if you believe that a question is ambiguous.*

Marks

Page	Max.	Achieved
2	**13**	
3	**14**	
4	**10**	
5	**8**	
6	**11**	
7	**10**	
Total	**66**	

1. {6 marks} Suppose we have 3 sets: A={1, 2, 3, 4, 5}, B={2, 4, 6, 8, 10}, and C={3, 5, 9, 10}. Find expressions, if they exist, which are equal to each of the following sets. These expressions must be expressed in terms of the set operators: union $\cup$, intersection $\cap$, and set difference $-$.

Example: {2, 3, 4, 5, 6, 8, 10}: This set can be represented by $(A \cap C) \cup B$.

Give such expressions for:

(a) {6, 8}:

(b) {2, 3, 4, 5, 9, 10}:

2. {7 marks} A security guard has m keys and n locks, but forgot which keys open which locks. (The keys are just for these locks.) Suppose that there are exactly 2 keys that open the same lock, while the rest of the keys open one and only one lock.

a) If $m = 6$ and $n = 5$, can the guard open *all* of the locks? Briefly justify your answer.

b) If $m = 6$ and $n = 5$, is this function one-to-one? Onto? No explanations are necessary.

c) Give the best Big-O estimate for the worst-case number of steps required to open all the locks, given m keys and n locks (where exactly two of the keys open one lock, and all other keys open only one lock). No witnesses are required, but briefly justify your answer (by showing your calculation).

3. {4 marks} Suppose Wendy's restaurants offer 8 toppings for a hamburger: cheese, ketchup, lettuce, mayonnaise, mustard, onions, pickles, and tomatoes. At Wendy's, you can tell the server at the counter which combination of toppings, if any, that you would like to have on your hamburger. What is the number of different ways that a hamburger can be "customized"? (i.e., How many different ways are there to come up with hamburgers having different toppings?) We assume that the order of toppings doesn't matter, and you're limited to one unit of each topping (i.e., someone can't say, "I want 4 pickles".)

4. {4 marks} Suppose that $f(n)$ is $\Omega(g(n))$, $h(n)$ is $\Omega(g(n))$, and $k(n)$ is $\Omega(g(n))$. Use the *definition* of Big-Ω to prove that $f(n) + h(n) + k(n)$ is $\Omega(g(n))$.

5. {6 marks}

a) How many 3-person committees can be chosen from a set of 6 people, if there are no restrictions as to who can be on the 3-person committee?

b) Suppose that 6 boys and 6 girls have signed up to be on a co-ed softball team. Suppose the rules state that only 7 players (from this group of 12) can play in a given game. For a given game, how many ways are there to select a team of 7 players if the rules also state that at least 3 of the 7 players must be girls?

6. {10 marks} The following C++ function traverses a doubly-linked list consisting of 0 or more nodes from the current node, in the direction indicated by the input parameter (going left, or going right). For example, if you are at node "Susan", and you're supposed to go left, then, on the screen, print out all the names found in the nodes from the one before Susan to the start of the list, in that order (but do not include Susan (in the current node). The same idea applies when going to the right.

The following Node code is correct:

```
struct Node {
    string  name;
    Node *  prev;
    Node *  next;
};
```

If `curr` is a Node pointer, then `cout << curr->name << endl;` prints out the name.

Finish the following function. There is no dummy node at the start or the end of the list, so you'll have to test for where the list ends. Note that the function returns the number of names printed.

```
int printNodesinOneDirection(Node * curr, char LeftOrRight)
{
  //  POST-CONDITION:  This function starts at node "curr" and then prints out
  //  all the names in the nodes to the left ('L') or right ('R') of curr (but
  //  excluding the name in curr.  The function returns a count of the number
  //  of names printed on the screen.
  //  PRE-CONDITION:  curr is NULL, or points to a valid Node shown above;
  //                  LeftOrRight is either 'L' or 'R'

  /* Declare any needed variables here.  Follow this with your code.  (Some code
     is provided further below.) */









  if (LeftOrRight == 'L')
    {
      /* print out all the names of the nodes to the left of curr's node */
```

More space is on the next page.

7. {4 marks} Which of the following statements about stacks, queues, and deques are true? Circle all correct answers.

a) Stacks and queues can be implemented using arrays, but deques cannot be.

b) A queue allows a user to retrieve nodes using the FIFO principle, in O(1) time.

c) A stack allows a user to retrieve nodes using the LIFO principle, in O(1) time.

d) Consider an array having a capacity of n elements, and assume that no more than one element can fit into a cell of the array. It is clear that we cannot store more than n elements in such an ordinary array. On the other hand, if the array is a *circular* array, then we can store more than n entries in it at any given time because we can make use of the modulus operator.

e) With a doubly-linked list, it is not possible to keep the nodes sorted, unless there is only 1 element in the list (or unless it's an empty list).

8. {0 marks} The women's Canadian Olympic hockey team did really well this year, winning a *gold* medal! The men's Canadian Olympic hockey team, however, didn't do very well, scoring exactly 0 goals in 3 of their last 4 games—and they received no medal. What was your opinion of the performance of the men's Canadian Olympic hockey team?

a) I'm disappointed, especially since the men collectively make $95 million per year in the NHL.

b) I'm downright angry.

c) I'm not at all surprised because _____.

d) The men did reasonably well, given the time constraints, pressure, etc. Besides, I expect that we'll strike gold at the 2010 Olympics in Vancouver.

e) I don't have an opinion (or I don't care, or I don't follow hockey).

9. {4 marks} Which of the following statements are *true* about complexity? Circle all correct answers.

a) Suppose you compute that $f(n)$ is $O(n)$, and that $f(n)$ is $\Omega(\lg n)$. Then it is the case that $f(n)$ is $\Theta(n)$.

b) For all comparison based sorts, an upper bound on the number of steps taken to sort n elements is $\Omega(n \lg n)$.

c) Insertion sort is a recursive algorithm, and it has worst-case complexity $\Theta(n^2)$.

d) Suppose Mergesort is used to sort an array A (containing unique elements) into ascending order. Then, a binary search on the result has complexity $\Theta(\lg n)$.

e) A $\Theta(n^2)$ algorithm is likely to outperform a $\Theta(n^{3/2})$ algorithm.

10. {6 marks} Compute Big-Theta for the number of dollar signs ($) printed by the following algorithm, as a function of *N*. Justify your answer by briefly explaining your work (i.e., by showing your calculations). It is *not* necessary to compute witnesses.

Algorithm MyCalc(N) written in C++ style:

```
j = 0;
n = N;

while (n > 1)
  {
    n = n / 32;
    j++;
  }

for (w = 0; w < (N/3 + 32); w++)
    for (k = 1; k <= j*N; k++)
        cout << "$";
cout << endl;
```

11. {5 marks} Give a recursive algorithm to compute the product of 2 positive integers *m* and *n* using only addition.

12. {4 marks}
How many non-negative integer solutions are there to the following equation?

$x_1 + x_2 + x_3 + x_4 + x_5 = 18$ such that $x_i \geq i$ (the letter "i"—*not* "one")

13. {6 marks} This question is about program correctness. Use a loop invariant to prove that the following C++ code function correctly computes x^n where $n \geq 1$. (You can assume that the result is an integer and will fit into an integer.)

```
int computeProduct(int x, int n)
{
   int product = 1;
   int k = 1;

   while (k <= n)
    {
      product = product * x;
      k++;
    }

   return product;
}
```