Name: _____

ID Number: _____

CSC 225 Midterm Exam
Oct. 14, 2010

Instructions:

1. Put your name on every page of the exam.
2. No calculators or other aids. Closed book.
3. Read through the entire exam before beginning. You should have 9 pages including this header page.

Question	Value	Mark
1	20	
2	20	
3	20	
4	20	
5	20	
Total	**100**	

Recall that you need at least 40% (40/100) in order to write the final exam in this course. Suggested strategy: read through the exam before starting, and begin with the questions which are easiest for you.

1. Consider the following sum: $S(n) = \sum_{i=1}^{n} i^5$.

(a) [5] Give a simple function $f(n)$ so that the sum $S(n)$ is in $\Theta(f(n))$.

(b) [5] State the definition of Big Oh.

(c) [5] Use your definition of Big Oh from (b) to prove that $S(n)$ is in $O(f(n))$ where $f(n)$ is your answer to part (a).

(d) [5] Prove that $S(n)$ is in $\Omega(f(n))$ where $f(n)$ is your answer to part (a).

2. [20] Solve the following recurrence using repeated substitution. Show all your work. Your final answer should be a closed formula.

$T(n) = n + 1 + 2\ T((n-1)/2), \quad T(3) = 5.$

You may assume that $n = 2^{k+1} - 1$ for some integer $k \geq 1$.

3. Consider this Java method:

```
public void readRear(Scanner in)
{    ListNode tmp, current; int data; int i;
     n= readInteger(in);
     start=null; rear=null;
     for (i=0; i < n; i++)
     {
          data= readInteger(in);
          tmp= new ListNode(data, null);
          if (i==0) { start=tmp; }
          else
          {
               current= start;
               while (current.next != null)
               {
                    current= current.next;  // Statement to count.
               }
               current.next= tmp;
          }
          rear= tmp;
     }
}
```

(a) [5] Set up a recurrence which counts the number of times that the statement with the comment **// Statement to count.** is executed for a given value of *n* and justify your formula.

[Question 3 continued]

(b) [5] Solve your recurrence from (a) to get a closed formula.

[Question 3 continued]

(c) [10] Prove by induction that your closed formula from (b) is the number of times that the given statement is executed for a problem of size n.

4. [20] Imagine that you are the teaching assistant for this class and give feedback regarding the efficiency and correctness of the pseudocode below for the splitList routine (the code looks a lot like C/Java but do not worry about syntax errors). Linked lists are as defined on our assignment having *n*, *start* and *rear* fields. Show how to correct each bug. Show how to fix the code to make it more efficient.

```
list1.n= size1;

list2.n= list.n - size1;

for (i=0; i <= size1; i++)
{

    list.start= list.start.next;

}

list1.rear= list.start;


list.start.next= null;


list2.start= list1.rear.next;

/* Find a pointer to the last cell on the second list. */

current= list2.start;

while (current.next != null)
{

     current= current.next;
}

list2.rear= current.next;
```

5. [20] Give recurrence relations that express the time complexities of the following algorithms that we have studied so far in this class. Include any assumptions you would make about n to make solving these recurrences manageable. Justify your recurrences, but you do not have to solve them.

(a) BeginMax.

(b) MiddleMax.

(c) Binary search in the worst case.

(d) Quicksort in the worst case.

(e) Quicksort in the best case.

Use this page if you need more space.
Clearly indicate the question you are answering.