

# CSC 172 MID-TERM EXAM

March 7<sup>th</sup>, 2008

Please write answers on the exam sheet.

Put your name and student number (not SS#) on the front (this) page.

Closed book, closed notes

Hand calculators are allowed.

Laptop computers are not allowed.

6 Questions

100 points total.

75 min

Meliora.

NAME: \_\_\_\_\_

QUESTION	GRADE	POINT VALUE
Q1		16
Q2		16
Q3		17
Q4		17
Q5		17
Q6		17
<b>TOTAL</b>		<b>100</b>

1. Prove the following relationship **by induction**. Be sure to explicitly state the induction parameter, the induction hypothesis and its use in the proof, and appropriately identify the salient cases.

$$\sum_{i=0}^n 2^i = 2^{n+1} - 1$$

2. An algorithm takes 10 ms for input size 1000. How long will it take for the input size 10000 (assuming that low-order terms are negligible) if the running time characterized by the following types. Most of the credit will be given for setting up the equations to describe the solution. So, you must show your work – don't just answer with a numerical time value.

a. Linear

b.  $O(n \log_{10} n)$

c. Quadratic

d. Exponential  $O(10^n)$

3. Show (prove) that the code below runs in  $(n^3 + 3n^2 + 2n)/6$  time, where  $n == a.length$ . To do this, you need two steps : (1) express the runtime as a series – relating lines in the code to expressions in the series (sigma notation) (2) show that the closed form of the series is equivalent to the expression above. *Note: direct proof is ok, proof by induction is not required, but may be used if you wish.*

```
public static int maxSubSum1( int [ ] a) {           //line #1
    int maxSum = 0;                                 // 2
    for( int i = 0; i < a.length; i++ )           // 3
        for( int j = i; j < a.length; j++ ){      // 4
            int thisSum = 0;                        // 5
            for( int k = i; k <= j; k++ )          // 6
                thisSum += a[ k ];                // 7
            if( thisSum > maxSum ){                // 8
                maxSum = thisSum;                 // 9
                seqStart = i;                      // 10
                seqEnd = j;                       // 11
            }                                      // 12
        }                                         // 13
    return maxSum;                                 // 14
}                                                 // 15
```

(extra page for Q#4)

4. Prove by structural induction that any binary tree with one or more nodes contains one fewer edges than it has nodes.

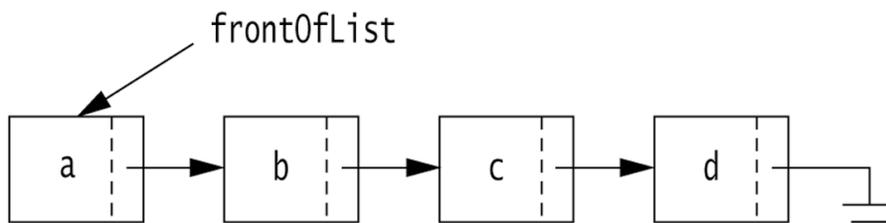
5. Given the recurrence  $T(1) = 1$ ,  $T(n) = T(n-1) + n$   
Derive (prove) the closed form of  $T(n)$

6. Show the result of inserting the following keys into a heap would work on the following insertion sequence {14,5,7,13,6,15,8,3,4}. Then do two deleteNext operations, then insert {12,2}.

(extra sheet)

7. Assume a “bare bones” implementation of a linked list

```
class Node {  
    Object data;  
    Node next;  
}
```



Write a JAVA method to print every other element in the list. Only a bare bones implementation is required, you need not use Generics or exception handling.

```
public void printAlternate(Node headOfList)
```

(extra sheet)