York University
COSC 3101 Fall 2004 – Midterm (Nov 4)
Instructor: Jeff Edmonds

Family Name: ____________________ Given Name: ____________________

Student #: ____________________ Email: ____________________

Section to which to return the midterm (circle one): A: TR 17:30, C: TR 11:30

Problem 1: (18 marks)	
Problem 2: (4 marks)	
Problem 3: (6 marks)	
Problem 4: (8 marks)	
Problem 5: (20 marks)	
Problem 6: (10 marks)	
Problem 7: (21 marks)	
Problem 8: (13 marks)	
Total (100 marks)	

This test is closed book and lasts 90 minutes. 0.9 minutes per mark.
You may not use any electronic/mechanical computation devices.
There are 7 pages including the cover page.

Keep your answers short and clear.

1. $2 \times 9 = 18$ marks: Compute the solution and tell which rules that you use.

   (a) $\sum_{i=0}^{n} 3^i \times i^8 = \Theta($
   Type of sum:

   (b) $\sum_{i=0}^{n} 300i^2 \log^9 i + 7\frac{i^3}{\log^2 i} + 16 = \Theta($
   Type of sum:

   (c) $\sum_{i=0}^{n} \sum_{j=0}^{n} i^2 j^3 = \Theta($

   (d) $7 \cdot 2^{3 \cdot n^5} = 3^{\Theta(n^5)}$ **True** **False**

   (e) What is the formal definition of $f(n) = n^{\Theta(1)}$?

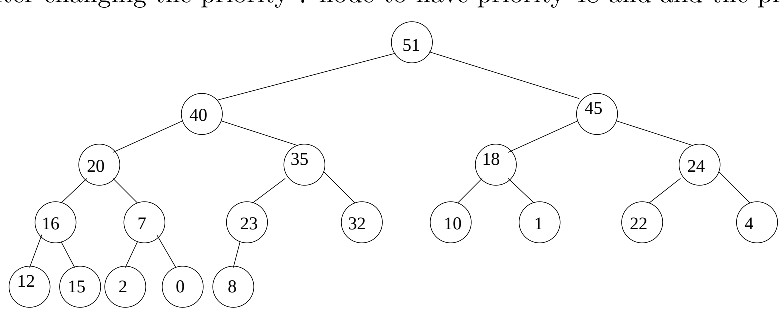
   (f) $n^2 - 100n \in o(n^2)$ **True** **False**

   (g) If $T(n) = 8T(\frac{n}{4}) + \Theta(n)$. Then $T(n) = \Theta($

   (h) The solution of the recurrence $T(n) = 5T(\frac{n}{\sqrt{5}}) + n^2 \log(n)$ is
   $T(n) = \Theta($

   (i) If $T(n) = 8T(n-5)$. Then $T(n) = \Theta($

2. 4 marks: Consider the following priority queue implemented as a heap. Consider a reasonable algorithm that changes the prioity of a node. (Its input includes a pointer to the node). Give the resulting heap after changing the priority 7 node to have priority 48 and and the priority 45 node to 3.



3. 6 marks: Considering Programs.

algorithm $Eg_1(n)$

$\langle$***pre−cond***$\rangle$**:** n is an integer.

$\langle$***post−cond***$\rangle$**:** Prints "Hi"s.

```
begin
        i = 0
        put "Hi"; put "Hi"; put "Hi"; put "Hi"
        loop i = 1 . . . n
                put "Hi"; put "Hi"; put "Hi"
                loop j = 1 . . . n
                        put "Hi"; put "Hi"
                end loop
        end loop
end algorithm
```

algorithm $Eg_2(n)$

$\langle$***pre−cond***$\rangle$**:** n is an integer.

$\langle$***post−cond***$\rangle$**:** Prints $T(n)$ "Hi"s.

```
begin
        if( n ≤ 1) then
                put "Hi"
        else
                put "Hi"; put "Hi"; put "Hi"
                Eg2(n/4)
                Eg2(n − 5)
        end if
end algorithm
```

(a) Give the exact time complexity (running time) of Eg_1.

(b) Give a recurrence relation for the running time of Eg_2. Do not solve it.

4. 8 marks: Briefly describe and contrast the difference between a "More of the Input" loop invariant and a "More of the Output" loop invariant. Give an example of each along with a picture.

5. 20 marks: Iterative Algorithms: You are now the professor. Which of the steps to develop an iterative algorithm did the student fail to do correctly in Eg_3? How?

algorithm $Eg_3(I)$

$\langle$***pre−cond***$\rangle$**:** I is an integer.

$\langle$***post−cond***$\rangle$**:** Outputs $\sum_{j=1}^{I} j$.

```
begin
        s = 0
        i = 1
        while( i < I )
            ⟨loop−invariant⟩: Each iteration adds the next
                term giving that s = ∑_{j=1}^{i} j.

            s = s + i
            i = i + 1
        end loop
return(s)
end algorithm
```

6. 10 marks: Consider my solution to Q3 in Assignment 2, which finds for each pair of nodes $u, v \in V$ of a graph a path from u to v with the smallest total weight from amongst those paths that contains exactly k edges.

   (a) 4 marks: What are the steps for maintaining the loop invariant.

   (b) 4 marks: What needs to be changed so that instead of the path having EXACTLY k edges, it has k or fewer edges?

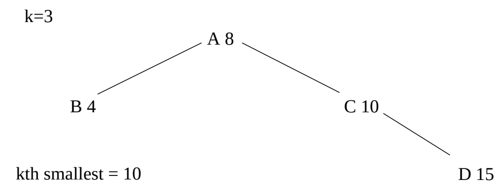
   (c) 2 marks: What needs to be changed so that for every u and v it outputs the shortest over all path. (Assume the weights are all positive.)

7. 21 marks: Recursion:

   (a) 4 marks: With Iterative algorithms, Jeff is obsessed with Loop Invariants. Describe in full the scenario that he is obsessed with regarding Recursion. (How is Recursion abstracted?)

   (b) 3 marks: Which are the general shapes of trees that you should check your program on. (Binary trees)

(c) 9 marks: Finding the $\boldsymbol{k^{th}}$ Smallest Element: Write a recursive program, which given a binary search tree and an integer $\boldsymbol{k}$ returns the $\boldsymbol{k^{th}}$ smallest element from the tree. (Code is fine.)



(d) 2 marks: Give the recurrence relation and the running time for your program when the input tree is completely balanced.

(e) 3 marks: Prove that no program can solve the problem by more than a constant factor faster.

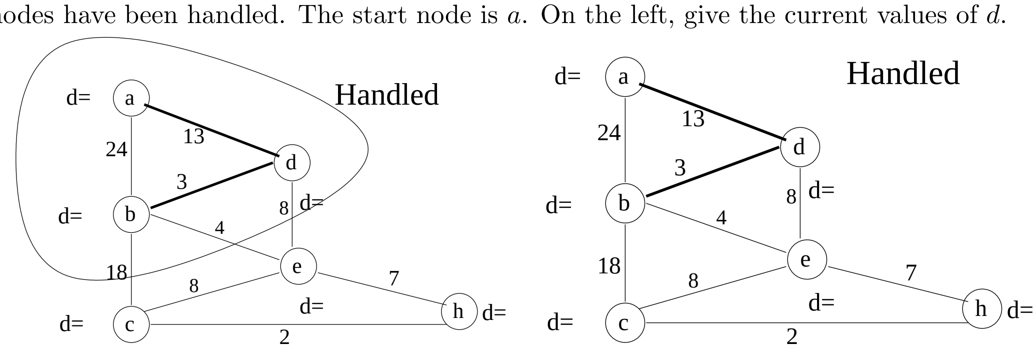
8. 13 marks: Dijkstra's Algorithm:

(a) 4 marks: Give the full loop invariant for Dijkstra's Algorithm. Include the definition of any terms you use.

(b) 2 marks: What is the exit condition for Dijkstra's Algorithm?

(c) 2 marks: Prove that the post condition is obtained.

(d) 3 marks: Consider a computation of Dijkstra's algorithm on the following graph when the circled nodes have been handled. The start node is a. On the left, give the current values of d.



(e) 2 marks: On the right, change the figure to take one step in Dijkstra's algorithm. Include as well any πs that change.