Quiz 1 — Solution

Engr 4892 Data Structures

13 July 2004

Total marks: 49

When answering complexity questions using big-Theta notation write the order of complexity as simply as possible; e.g. $\Theta(N)$ *rather than* $\Theta(5N+3)$.

Q0 [9]

Suppose programs p, q, and r have worst-case time functions

$$\begin{aligned}
WT_p(N) &= 17 \qquad \text{nanoseconds} \\
WT_q(N) &= 7\log_2 N + 20 \qquad \text{nanoseconds} \\
WT_r(N) &= 59N\log_2 N + 42N\log_{10} N + 30N + 12\log_2 N + 3 \qquad \text{nanoseconds}
\end{aligned}$$

Express their orders of complexity *as simply as possible* using big-Theta notation:

p: $\Theta(1)$

q: $\Theta(\log N)$

r: $\Theta(N \log N)$

Q1 [10]

Design a *recursive* subroutine that prints a positive number in base 8 to output stream cout.. (Recall that in C++, i/8 and i%8 compute the quotient and the remainder, respectively, of i divided by 8.)

```
void printBase8( unsigned i ) {
    if( i >= 8 )
        printBase8( i/8 ) ;
    cout << i%8 ;
}
```

Another possibility

```
void printBase8( unsigned i ) {
    if( i & ~7 )
        printBase8( i >> 3 ) ;
    cout << i & 7 ;
}
```

Q2 [10]

Suppose that you are given a boolean query represented as in the last assignment. A query written in “functional notation” looks like this:

```
OR(AND("Ginger", NOT(AND("Spice", "Girls"))), "Garlic")
```

The following declaration has been added to the four classes derived from QueryNode.

```
virtual void printFN( ) ;
```

Design a set of *recursive* subroutines to print a query in functional notation to output stream cout.

You must supply the 4 definitions:

```
void StringNode::printFN() {
    cout << '"' << string_ << '"' ;
}
void AndNode::printFN() {
    cout << "AND(" ;
    leftChild_ -> printFN() ;
    cout << ", " ;
    rightChild_ -> printFN() ;
    cout << ")" ;
}
void OrNode::printFN() {
    cout << "OR(" ;
    leftChild_ -> printFN() ;
    cout << ", " ;
    rightChild_ -> printFN() ;
    cout << ")" ;
}
void NotNode::printFN() {
    cout << "NOT(" ;
    child_ -> printFN() ;
    cout << ")" ;
}
```

Q3[8]

(a) For the subroutine in question 1, what are the stopping condition and a variant?

Stopping Condition: $i < 8$

Variant: i *[Another valid answer would be $i/8$]*

(b) For the group of subroutines in question 2, what is the stopping condition and a variant?

Stopping Condition: **this is a StringNode*

Variant: *The height of the query. [Another valid answer would be the size of the query.]*

Q4[6] (Answer with big-Theta notation.)

(a) For the subroutine in question 1, what is the time complexity, in terms of the value of parameter i? (You should assume that each call to a library routine has a time complexity of $\Theta(1)$).

Complexity: $\Theta(\log i)$

(b) For the group of subroutines in question 2: Let S be the size of the query in terms of nodes and H be the height of the query. What is the time complexity of printFN, in terms of S and H. (You should assume that each call to a library routine has a time complexity of $\Theta(1)$).

Complexity: $\Theta(S)$ *[Another valid answer would be $\Theta(2^H)$]*

[Surprisingly few people did well on this question. The reason for $\Theta(S)$ is that a constant amount of work needs to be done for each node of the tree. The reason for $\Theta(\log i)$ is that there is a constant amount of work done for each 3 bits of the number. The number of bits required to represent a number is the log base 2 of the number.]

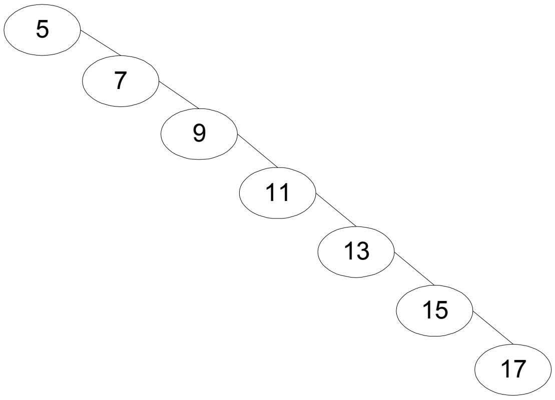
[Marking note: A number of students answered in terms of N. This is not useful unless I know what quantity N represents. I don't think I took off any marks for this this time.]

Q5[6] Assume numerical ordering is used for labels.

(a) Start with an empty labeled binary search tree. Insert nodes in the following sequence

$$\langle 5, 7, 9, 11, 13, 15, 17 \rangle$$

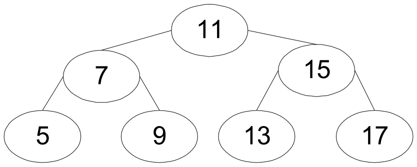
using the algorithm given in class. Draw the final tree



(b) Start with an empty labeled binary search tree. Insert nodes in the following sequence

$$\langle 11, 7, 5, 9, 15, 13, 17 \rangle$$

using the algorithm given in class. Draw the final tree


11
7
15
5
9
13
17