

Name _____

CSC-477/677: Algorithm Analysis
Fall 2014
Homework #2
Due on: *September 16, 2014*

Q1. Solve the following recurrences.

a) $F(n) = F(n/3) + 9$

A:

b) $G(n) = 8G(n/2) + n^3$

A:

c) $H(n) = H(n/2) + n^2$

A:

d) $T(n) = T(\sqrt{n}) + 1$

A:

e) $F(n) = F(n-2) + 5n^2$

A:

Q2. Determine the asymptotic time complexity, in terms of n , of each of these code fragments.

a)

```
for(int i = 1; i < n; i = i*2)
  for(int j = 1; j < i; j++)
    cout << "Hello world" << endl;
```

A:

b)

```
for(int i = 1; i < n; i = i*2)
  for(int j = 1; j < n; j = j++)
    cout << "Hello world" << endl;
```

A:

c)

```
for(int i = n; i > 1; i = sqrt(i))
  cout << "Hello world" << endl;
```

A:

Name _____

```
d) for(int i = n; i > 1; i = log(i))  
    cout << "Hello world" << endl;
```

A:

Q3. Consider the following procedure (it is not written in any particular programming language).

```
procedure q(int n) {  
    if (n > 0) {  
        int m = n/2;  
        q(m);  
        q(m-1);  
        q(m-2);  
        q(m-3);  
    }  
}
```

You need to analyze the time complexity of this procedure, in terms of the parameter **n**.

(a) What is the asymptotic time complexity of this procedure, in terms of the parameter **n**? Give an argument to support your answer.

A:

(b) Write code for this procedure, in any programming language you wish. Run the procedure for various values of **n**, including 10, 100, 1000, and 10000. Turn in a graph summarizing the results of your experiment, and compare your results to your answer in part **(a)**.

A:

Name _____

Q4. This problem deals with Euclid's algorithm for finding the greatest common divisor of two non-negative integers.

Suppose that **n** and **m** are two positive integers, and that **m** < **n**. How many steps does it take Euclid's algorithm to compute the greatest common divisor of **n** and **m**? I want the answer expressed asymptotically in terms of **n**, for example, **O(n)** or **Theta(n)**, or **O(log n)**, or whatever.

You may assume that it takes one time step to do any operation, such as **mod**, or to do an assignment.

You must give the sharpest possible answer. If the correct answer is **Theta(n)** and you write **O(n)**, for example, you will lose points, even though technically your answer is correct, because it is not the sharpest possible answer.

Do not give a proof that your answer is correct.