Fundamentals of Algorithms
CS502-Fall2011

ASSIGNMENT1

Deadline

Your assignment must be uploaded/submitted at or before **1st November 2011.**

Uploading instructions

Please view the **assignment submission process** document provided to you by the Virtual University to upload the assignment.

Rules for Marking

It should be clear that your assignment will not get any credit if:

- The assignment is submitted after due date.
- The submitted assignment does not compile or run.
- **The assignment is copied.**

Objectives

This assignment will help you to understand the concepts of Asymptotic Growth, making analysis of pseudo code, recurrence relation development, asymptotic function understanding and iterative solutions for recurrences.

Guidelines

RULES FOR CALCULATING TIME COMPLEXITY AND BIG-OH

Rule 00
Normally these formulas are very handy:

If $x^y = z$ then $y = \log_x z$
Also,

$$\sum_{i=1}^{n} a_i = \frac{n}{2}(a_1 + a_n) \qquad \sum_{i=1}^{n} i = \frac{n}{2}(n+1) \qquad \sum_{k=0}^{m} r^k = \frac{1-r^{m+1}}{1-r}$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \text{ (for n >=1)}$$

Rule 0

The condition that stops a loop executes ONE MORE time than the loop itself (the last time is when it is evaluated false)

Rule 1

for (i=0;i<n;i=i+k) Anything inside the loop will run approximately **n/k** times

Rule 2

for (i=n;i>0;i=i-k) Anything inside the loop will run approximately **n/k** times

Rule 3

for (i=1;i<n;i=i*k) Anything inside the loop will run approximately **$\log_k$n** times

Rule 4

```
for(i=1;i<=n;++i)
        for (j=1;j<=i;++j)
```

The above nested loop approximately runs **½ n(n+1)** times.
The variable j depends upon the value of i

Rule 5

```
for(i=1;i<=n;i=i*2)
        for (j=1;j<=i;++j)
```

The statements in the above nested loop approximately run **2n-1** times.
The variable j depends upon the value of i

Rule 6

If the loop variables are independent then the total times a statement inside a nested loop is executed is equal to the product of the times the individual loops run

e.g.

```
for (i=0;i<n;++i)
        for (j=0;j<m;++j)
```

A statement inside the above nested loop will run **n*m** times

Other Guidelines

While loop related information

Complexity of “while” loop depend upon the initial entrance condition if it remains true for “n” iterations it will be “n+1”; Note here “1” will be added for the last time check of the condition .Here this will be clear to you if the some logical conditions are checked other then counters then all complexity will be based on scenario of the problem and nature of the logical condition.

Function Growth rate concept

If some function $f_1(x)>f_2(x)$ for positive values of x then the function $f_1(x)$ is said to have greater growth rate then $f_2(x)$. For example $f_1(x)=x^5$ and $f_2(x)= x^6$ it is obvious that $f_2(x)$ has greater growth rate ($2^6 > 2^5$).This concept relate to complexity of algorithm ,an algorithm having greater growth rate function means the algorithm has greater complexity here $f_2(x)$ is more complex then $f_1(x)$.

Estimated Time 2.5 hour

For all parts of the question to understand maximum time is 1.25 hours and for solution maximum time is 1.25 hour. It all depends upon your sheer concentration.

Question **(4*5)**

Write piece of pseudo codes having the following time complexities:

a) $T_1(n) = n + \sum_{i=0}^{n} i$

b) $T_2(n) = 6n(\log_{12} n) + 1 + (m/7)\log(m)$

c) **O (z+m)**

d) $T_3(n) = 5n\sum_{i=1}^{n}(1/i)$

Note: **Where the base of "log "has not been mentioned take it as base "2".**