Fundamentals of Algorithms
CS502-Fall2014

ASSIGNMENT 1

Deadline

Your assignment must be uploaded/submitted at or before **20th November 2014.**

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 is to revise basic level concepts to make basis for advance concepts of Analysis of Algorithms. This assignment will help you to understand the concepts of Asymptotic Growth, making analysis of pseudo code, asymptotic function understanding and iterative solutions for recurrences. You can write the pseudo codes form given functions.

Guidelines

RULES FOR CALCULATING TIME COMPLEXITY AND BIG-O

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 3 hours

For both questions to understand; maximum time is 1.50 hours and for solution maximum time is 1.50 hours. It all depends upon your sheer and devoted concentration.

Special notes:

To solve the assignment you should have grip on the delivered lessons and also consult above guidelines to perceive completely.

Your own effort is appreciated and you will rewarded…

Note:Those stuents who will Copy and pate assignment from internet or from any other source as its will be rewarded as F. in this course …

Question 1 (10)

Find the running time complexity of the following piece of code and show your working step by step.

```
 yz =0;
  xw=0;
 for(i=m; i>-6; i=i-6)
 { xw++; }

 for (i=n; i>-2015;i=i-5)
 { yz=yz+1;}

 for (i=1;i<=n;i=i*5)
 {        for (j=1;j<=5n;j*12
          {
              for(k=n;k>-5; k=k-4)
                   {
                    x=x+12
                    }
          }
}

Print x;

While(k<=z)
{
k=k*2
(
for(m=k; m>=-100000; m=m-1
)
}
Print k;
```

Question 2 (10)

Arrange the following in the Most to Least complexity order. Here "n "is processing steps for some complexity functions and j< k and j & k are numbers greater than 2. Every function is separated by "comma". Note: These functions must be arranged on generic basis. Further there are 20 functions to arrange and each line has five functions.

$n/1000000,\ n^{\sqrt{k}}/1000,\ n^{Log_2\ j/2},\ 1000nLog_3 n,\ n^{n-1000},$

$1000000000000,\ \frac{n\log_2 n}{10000},\ 2^{1000n},\ \frac{1}{1000}Log_2(\sqrt{n}),\ 100000n!,$

$(n!nLog_4 n),\ n/\sqrt[12]{n^2},\ n!nLog_2 n/\sqrt{n^2},\ 10000Log_2(Log_3 n),\ 1000n\ /\sqrt[5]{n},$

$n(Log_7 n)\sqrt[8]{n},\ \frac{1}{2150}n\sqrt[5]{j},\ n(Log_8 n)\sqrt[9]{n},\ 100000Log_2 n,\ 2199n^{Log_5\ j/2}$

Think Beyond The Boundaries. First Food for Thought….

All the Best!