

CS-2009: Design and Analysis of Algorithms

Assignment # 1

Submission Guideline:

- Your answers should be handwritten. You have to submit your assignment in both hard copy and soft copy.
- Mention your Name, Roll no and Section on front page.
- Submit your hard copy in A209F or A209E till 3pm (Thursday 12 Sep).
- You have to submit soft copy on GCR.
- Use camscanner application to compile your assignment in a pdf format in a single file. Name it as your name and roll no and submit on GCR in due time.
- No late submissions will be accepted.

Q no.1 Calculate time complexity functions $F(n)$ and Big oh of the following code snippets. Assume there are no errors.

(A)

```

int Function (Array A<int>)
{
    int i=1,j=1;
    int f=0,g=0;

    while (i<=|A|){

        if(A[i]==A[j])
            g=g+1;

        j=j+1;
        if(j>|A|){

            if(g>f)
                f=g;

            g=0;
            i=i+1;
            j=1 ;

        }
    }
    return f;
}

```

(B)

```
for i = 1 to n do  
  
    for j = 1 to 2 * i + 1 do  
  
        print ("Hello World")  
  
    end for  
  
end for
```

(C)

```
for i = 1 to n * n do  
  
    for j = 1 to i do  
  
        print ("Hello World")  
  
    end for  
  
end for
```

(D)

```
for (i = 0 to m) do
```

```
    t  $\leftarrow$  1
```

```
    while (t < m) do
```

```
        print ("Hello world")
```

```
        t  $\leftarrow$  t * 2
```

```
    end while
```

```
end for
```

(E)

```
(c) sum = 0;  
    for(i=1;i<2*n;i++)  
        for(j=1;j<i*i;j++)  
            for(k=1;k<j;k++)  
                if (j % i == 1)  
                    sum++;
```

(F)

```
procedure MatrixMultiplication(A, B)
  input A, B n*n matrix
  output C, n*n matrix
```

```
begin
  for ( i = 0; i < n; i++)
    for ( j = 0; j < n; j++)
      C[i,j] = 0;
    end for
  end for

  for ( i = 0; i < n; i++)
    for ( j = 0; j < n; j++)
      for( k = 0; k < n; k++)
        C[i,j] = C[i,j] + A[i,k] * B[k,j]
      end for
    end for
  end for
end MatrixMultiplication
```

Q no.2

- a) Prove that $f(n) = 100n + 5 \neq \Omega(n^2)$
- b) Find Theta bound for $f(n) = n^2/2 - n/2$
- c) Prove that $6n^3 \neq \theta(n^2)$
- d) Prove $f(n) = n \neq \theta(\log n)$

e) Do $f(n) = \log_2 n^3$ belongs to Big oh $O(\log_2 n)$? Prove your answer.

Q no.3

Suppose you are going to distribute plate of a Biryani on the road. You will ask people to get in a line if they want biriyani. There is no way to control that each person will get exactly one plate. People can rejoin in the line to get multiple plates. Assume that each person is assigned unique positive integer Id number.

You will keep distributing plates unless line is empty. Any number of people can join your line. Assume at the time of distribution, you can view Integer id of a person who is collecting current plate.

Now write a pseudocode which can show ID of a person who collected highest number of the biriyani plates that is more than $n/2$ of the total plates.

Note: (Your algorithm's time complexity growth should not be more than linear curve)