Lab Assignment 5: Time Complexity

Instructions:

1. You have to solve the assignment using pen and paper. Build a pdf file of the scanned copy of your written assignment using camscanner or any other App.
2. Name of the pdf should be written as yourfullname_enrollmentnumber.pdf

Deadline: 18th April 2020

Total Marks: 22

Late Evaluation: 20% penalty if delayed by one week. 50% penalty if delayed by two week. No submission will be entertained after that.

1. Find the complexity of the following program in terms of Big o notation. Explain the procedure also. **(3 Marks)**

(a)
```
Int main()
{
int i, j, k, n;
i=n;
while(i>1)
{ j=i;
   while(j<n)
   { k=0;
      while(k<n)
       {k=k+2;
        }
     j=j*2;
    }
 i=i/2;
}
```

(b)
```
Void fun(int n)
{
  int i=1, s=1;
  while(s<=n)
   {
      i++;
      s=s+i;
      printf("hi");
   }
}
```

(c)
```
Void fun(int n)
{
  int i, count=0;
  for(i=0; i*i<=n;i++)
     count++;
}
```

(d)
```
Void fun(int n)
{
  int i, j;
  if(n==1)
```

```
  return;
  for(i=0; i<=n;i++)
   {
      for(j=0; j<=n;j++)
       {
          printf("hi");
          break;
      }
    }
}
```

(e)

```
Void fun(int n)
{
  int i, j;
  for(i=1; i<=n;i++)
   {
      for(j=1; j<=n;j+=i)
       {
          printf("hi");
      }
    }
}
```

(f)

```
Void fun(int n)
{
  int i, count;
  if(n<2)
  return;
  else
  count=0;
  for(i=0; i<8;i++)
     fun(n/2);
  for(i=0; i<n^3;i++)
    count=cont+1;
}
```

2. Find the complexity of following recurrence relations using relevant formula. **(3 Marks)**

(a) $T(n)=T(n-1) + n$

(b) $T(n) = T(n/3-2) + n/2$

(c) $T(n) = 2T(n/4) + \sqrt{n}$

(d) $T(n) = T(n-2) + n^3$

(e) $T(n) = 7T (n/2) + n^2 \log n$

(f) $T(n) = T (7n/10) + n$

3. Find the complexity of the recurrence: $T(n) = 2T (n^{1/2}) + \log n$ **(1 Marks)**

4. Find the complexity of following relation without using any formula. Write the whole procedure to calculate the complexity. **(6 Marks)**

(a) $T(n) = T(n/2) + n\log n$

(b) $T(n) = T(n-1) + \log n$

(c) $T(n) = T(n-2) + 1/\log n$

(d) $T(n) = T(n/2) + T(n/4)+T(n/8) +n$

(e) $T(n)=T(n-a)+T(a) + cn$, where $a>=1$, $c>0$

(f) $T(n) = T(n-1) + T(n/2) + n$

5. If $f(n)= 3n^{\sqrt{n}}$ and $g(n)= 2^{\sqrt{n}\log n}$ then prove that $f(n)=O(g(n))$ **(1 Marks)**

6. If $T(n)=aT(n/b) + O(n^k)$ then derive the Master Theorem formula. Also, analyse that why a/b^k is an important factor while deriving the formula for Master Theorem. Also, analyse that how the behavior of block diagram depends on a/b^k. **(3 Marks)**

7. Find the complexity of the following programs in terms of Big O notations. Explain the procedure also. **(3 Marks)**

(a) Binary Search (b) Bubble Sort (c) Function to calculate n^{th} Fibonacci number

8. Write short note on following: **(2 Marks)**

(a) Run time analysis (b) Rate of growth (c) Best case, average case, and worst case (d) Big O, Big Θ, and Big Ω.