Assignment - I

Time Complexity

Question 1. Arrange these in the increasing order of time complexity:

```
f1(n) = 2^n
f2(n) = n^(3/2)
f3(n) = nlog(n)
f4(n) = n^(logn)
```

Question 2. Find both the best case and the worst-case Time complexity for the given code snippets. Assume all inputs are already fed into the arrays and variables.

(a)

```
int n, arr[n], k;

for(int i=0; i<n; i++)
{
    if( arr[i] == k )
        break;
}
```

(b)

```
int i,j,n;

for( i=0; i<=n; i++)
{
    for( j=0; j<log(i); j++)
    {
       cout<<"HI";
    }
}
```

(c)

```
int n, i, j, count = 0;

for( i = 1; i<=n; i++ )
{
     for( j = 1; j*j <= n; j++ )
     {
            count++;
     }
}
```

(d)

```
int pow(int a, int n)
{
    int sum=0;
    if( a == 0 )
    {
        return 1;
    }
    else if( n == 1 )
    {
        while(a--)
        {
            sum++;
        }
        return sum;
    }
    else
    {
        while(a--)
        {
            sum++;
        }
        return sum*pow(a,n-1);
    }
}
```

(e)

```
int count = 0;

for (int i = N; i > 0; i /= 2)
{
        for (int j = 0; j < i; j++)
                count += 1;
}
```

(f) Euclidean GCD algorithm

```
long long gcd(long long a, long long b)
{
   if(b>a)
      return gcd(b, a);

   if(b == 0)
      return a;
   else
      return gcd(b, a%b);
}
```

(g)

```
int theLast(int x, int y)
{
    if( x == y )
        return x+y;
    else if( x > y )
    {
        while( x > y )
            x--;

        return theLast(x,y);
    }
    else
    {
        while( y > x )
            y--;

        return theLast(x,y);
    }
}
```

(h)

```
int max, sum[max]={0},i,j;

for( i=1; i<max; i++)
{
     for( j=i; j<max; j+=i )
          sum[j]+=i;
}
```

(i)

```
int n, i, j, count = 0;

for( i = 1; i<=n; i++ )
{
   for( j=1; j*i <= n; j++ )
         count++;
}
```

(j)

```
int fib(int n)
{
   if(n==0)
         return 1;
   else
         return(fib(n-1) + fib(n-2));
}
```