DESIGN AND ANALYSIS OF ALGORITHMS LAB3

Name:V.Prasanna Vyshnavi

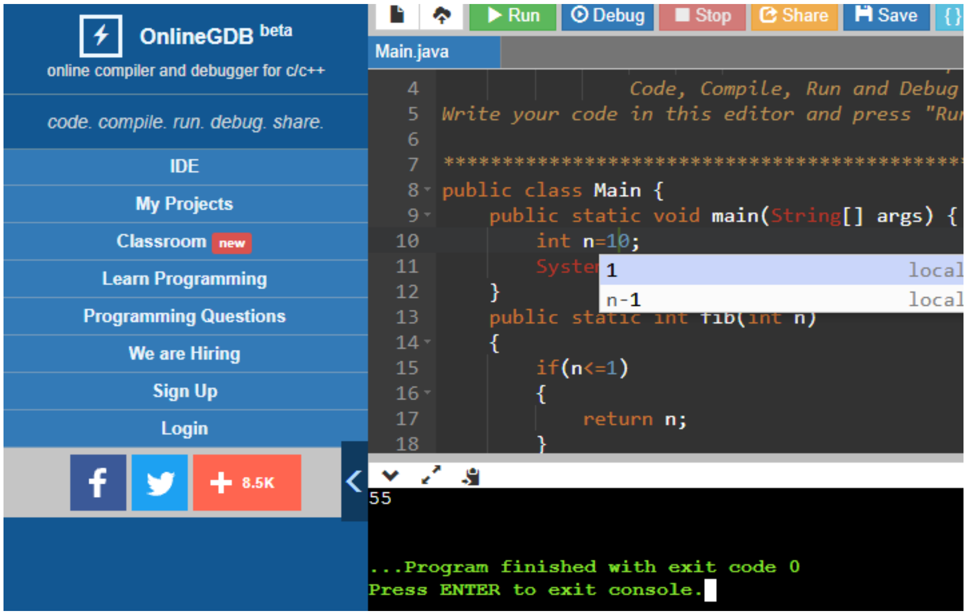
Regno:19BCE7661

Naïve Fibonacci series:

Code:

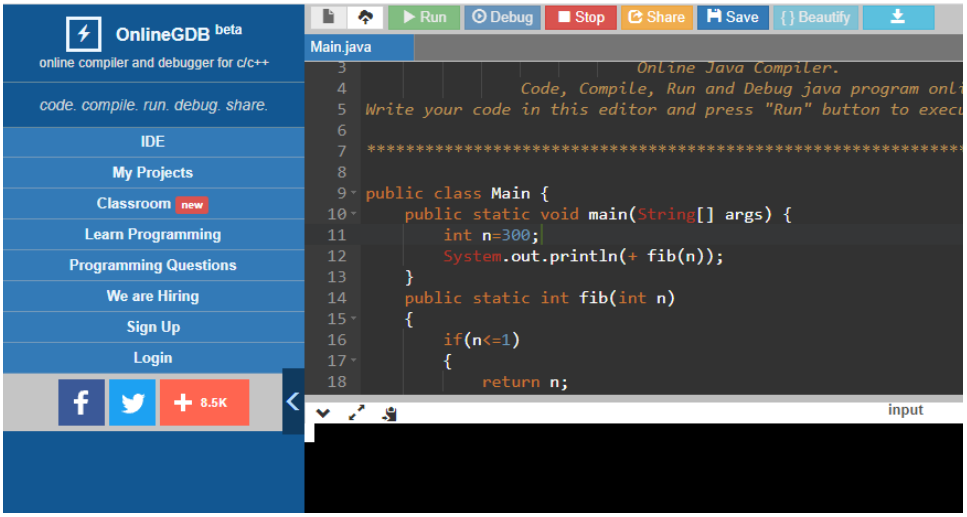
```
public class Main {

    public static void main(String[] args) {
        int n=300;
        System.out.println(+ fib(n));
    }
    public static int fib(int n)
    {
        if(n<=1)
        {
            return n;
        }
        else
        {
            return fib(n-1)+fib(n-2);
        }
    }

}
```

Output:



```
                Code, Compile, Run and Debug
Write your code in this editor and press "Ru

*********************************************
public class Main {
    public static void main(String[] args) {
        int n=10;
        Syste 1                             local
          n-1                               local
    }
    public static int fib(int n)
    {
        if(n<=1)
        {
            return n;
        }
```

```
55


...Program finished with exit code 0
Press ENTER to exit console.
```

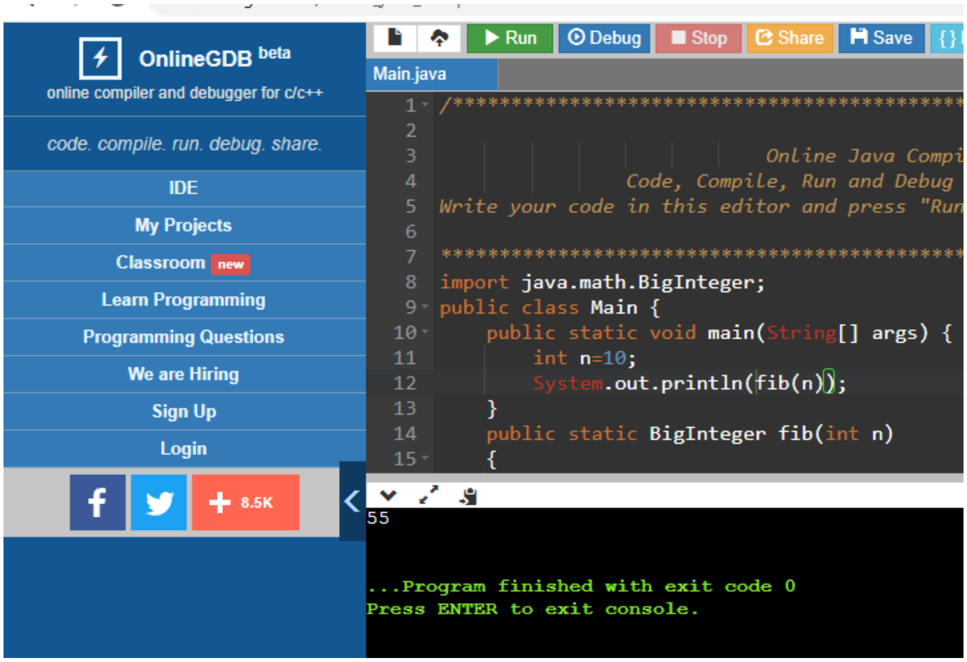


```
                          Online Java Compiler.
                Code, Compile, Run and Debug java program onl
Write your code in this editor and press "Run" button to execu

***************************************************************

public class Main {
    public static void main(String[] args) {
        int n=300;
        System.out.println(+ fib(n));
    }
    public static int fib(int n)
    {
        if(n<=1)
        {
            return n;
```

Effective Fibonacci series:

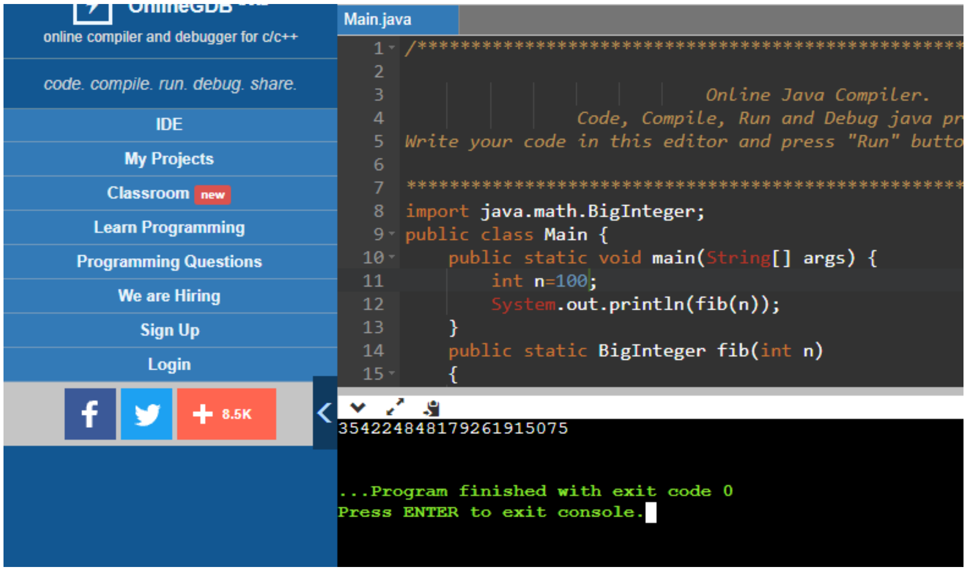
```
import java.math.BigInteger;
public class Main {
    public static void main(String[] args) {
        int n=100;
        System.out.println(fib(n));
    }
    public static BigInteger fib(int n)
    {
        BigInteger a=BigInteger.valueOf(0);
        BigInteger b=BigInteger.valueOf(1);
        BigInteger c=BigInteger.valueOf(1);
        for(int i=2;i<=n;i++)
        {
            c=a.add(b);
            a=b;
            b=c;
        }
       return (b);
    }

}
```

Output:



```
/***********************************************
                              Online Java Compi
                        Code, Compile, Run and Debug
Write your code in this editor and press "Run

***********************************************
import java.math.BigInteger;
public class Main {
    public static void main(String[] args) {
        int n=10;
        System.out.println(fib(n));
    }
    public static BigInteger fib(int n)
    {
```

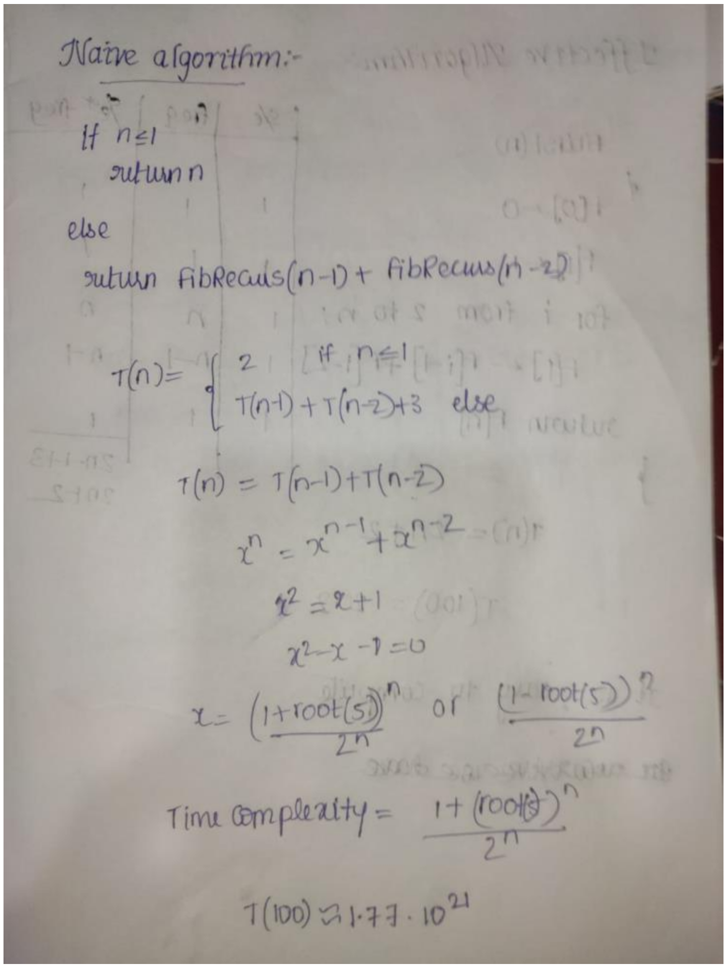
```
55


...Program finished with exit code 0
Press ENTER to exit console.
```



```
/**************************************************
                              Online Java Compiler.
                        Code, Compile, Run and Debug java pr
Write your code in this editor and press "Run" butto

**************************************************
import java.math.BigInteger;
public class Main {
    public static void main(String[] args) {
        int n=100;
        System.out.println(fib(n));
    }
    public static BigInteger fib(int n)
    {
```

```
354224848179261915075


...Program finished with exit code 0
Press ENTER to exit console.
```

Asymptotic Analysis:



Naive algorithm:-

```
if $n \le 1$
    return n
else
    return fibRecurs(n-1) + fibRecurs(n-2)
```

$$T(n) = \begin{cases} 2 & \text{if } n \le 1 \\ T(n-1) + T(n-2) + 3 & \text{else} \end{cases}$$

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

$$x^n = x^{n-1} + x^{n-2}$$

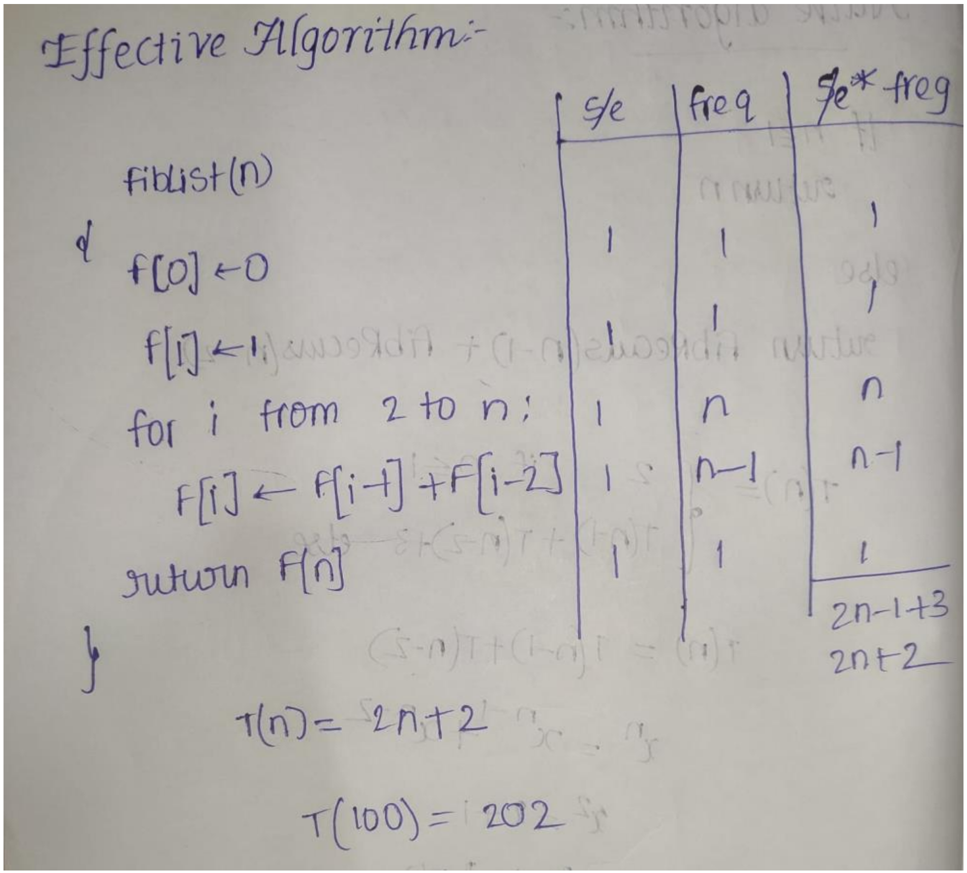
$$x^2 = x + 1$$

$$x^2 - x - 1 = 0$$

$$x = \frac{(1 + root(5))^n}{2^n} \text{ or } \frac{(1 - root(5))^n}{2^n}$$

$$\text{Time complexity} = \frac{1 + (root(5))^n}{2^n}$$

$$T(100) \approx 1.77 \cdot 10^{21}$$

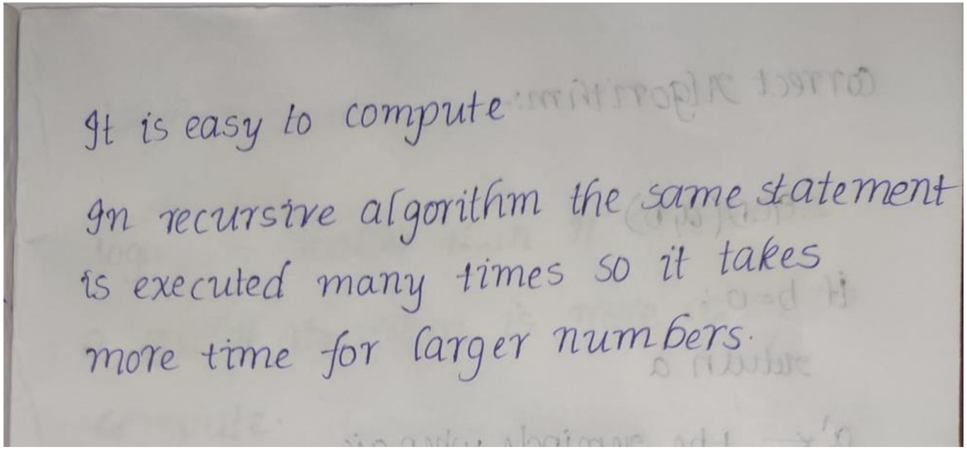


Effective Algorithm:-

	s/e	freq	s/e * freq
fiblist(n)			
{			
f[0] ← 0	1	1	1
f[1] ← 1	1	1	1
for i from 2 to n:	1	n	n
f[i] ← f[i-1] + f[i-2]	1	n-1	n-1
return f[n]	1	1	1
}			2n-1+3 2n+2

$T(n) = 2n+2$

$T(100) = 202$



It is easy to compute

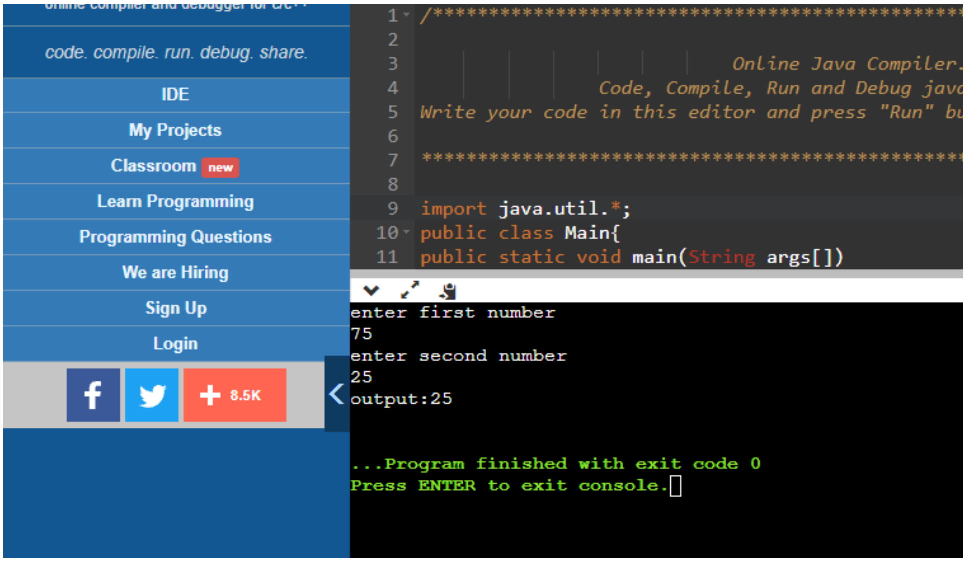
In recursive algorithm the same statement is executed many times so it takes more time for larger numbers.

Naïve GCD:

```
import java.util.*;
public class Main{
public static void main(String args[])
{
```

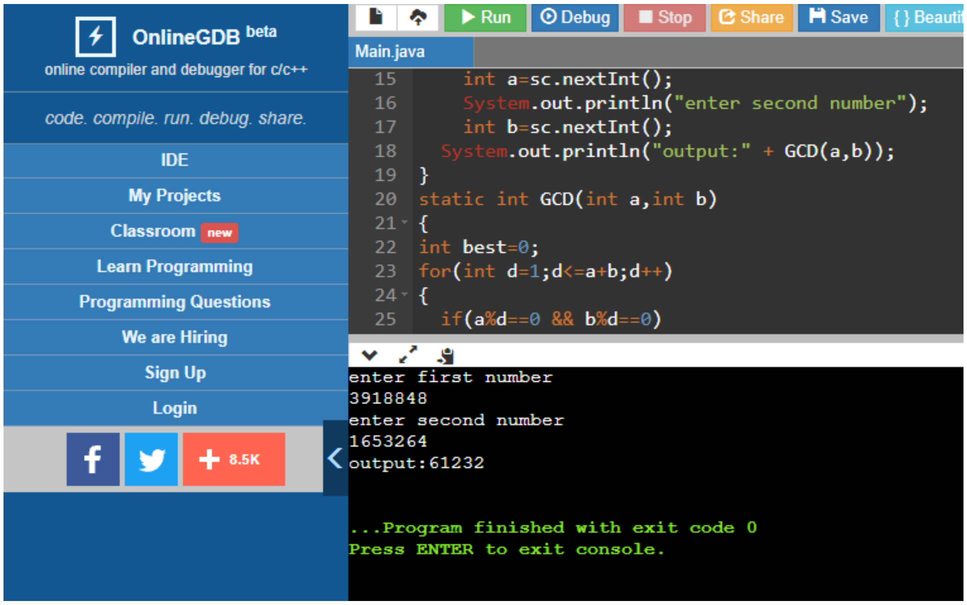
```
Scanner sc=new Scanner(System.in);
System.out.println("enter first number");
int a=sc.nextInt();
System.out.println("enter second number");
int b=sc.nextInt();
  System.out.println("output:" + GCD(a,b));
}
static int GCD(int a,int b)
{
int best=0;
for(int d=1;d<=a+b;d++)
{
  if(a%d==0 && b%d==0)
  {
  best=d;
  }
}return best;
}
}
```

Output:



```
/**********************************************
                              Online Java Compiler.
                Code, Compile, Run and Debug java
Write your code in this editor and press "Run" bu

***********************************************

import java.util.*;
public class Main{
public static void main(String args[])
```

```
enter first number
75
enter second number
25
output:25


...Program finished with exit code 0
Press ENTER to exit console.
```



```
    int a=sc.nextInt();
    System.out.println("enter second number");
    int b=sc.nextInt();
  System.out.println("output:" + GCD(a,b));
}
static int GCD(int a,int b)
{
int best=0;
for(int d=1;d<=a+b;d++)
{
  if(a%d==0 && b%d==0)
```

```
enter first number
3918848
enter second number
1653264
output:61232


...Program finished with exit code 0
Press ENTER to exit console.
```

Euclidian GCD:

import java.util.*;

public class Main{

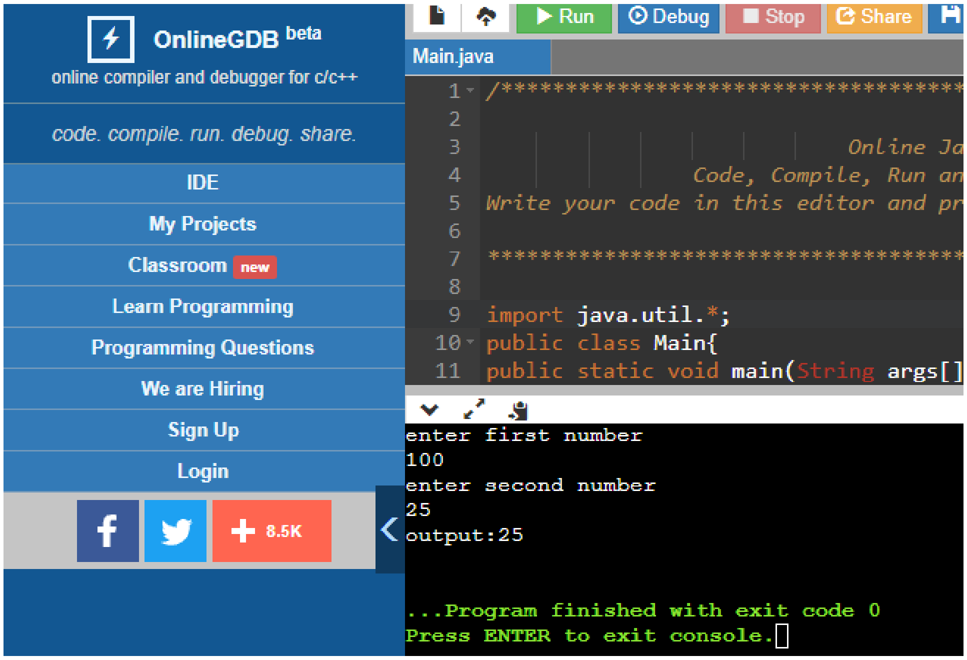
public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

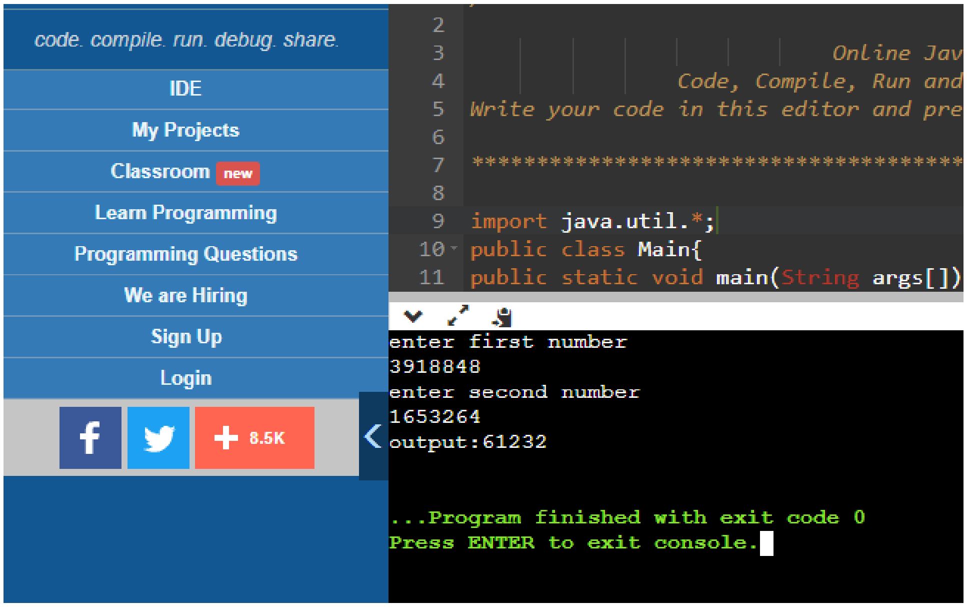
```
System.out.println("enter first number");
int a=sc.nextInt();
System.out.println("enter second number");
int b=sc.nextInt();
  System.out.println("output:" + euclidGCD(a,b));
}
static int euclidGCD(int a,int b)
{
if(b==0)
{
return a;
}
int a1;
a1=a%b;
return euclidGCD(b,a1);
}
}
```

Output:



```
/************************************
                                    Online Ja
                              Code, Compile, Run an
Write your code in this editor and pr

*************************************

import java.util.*;
public class Main{
public static void main(String args[]
```

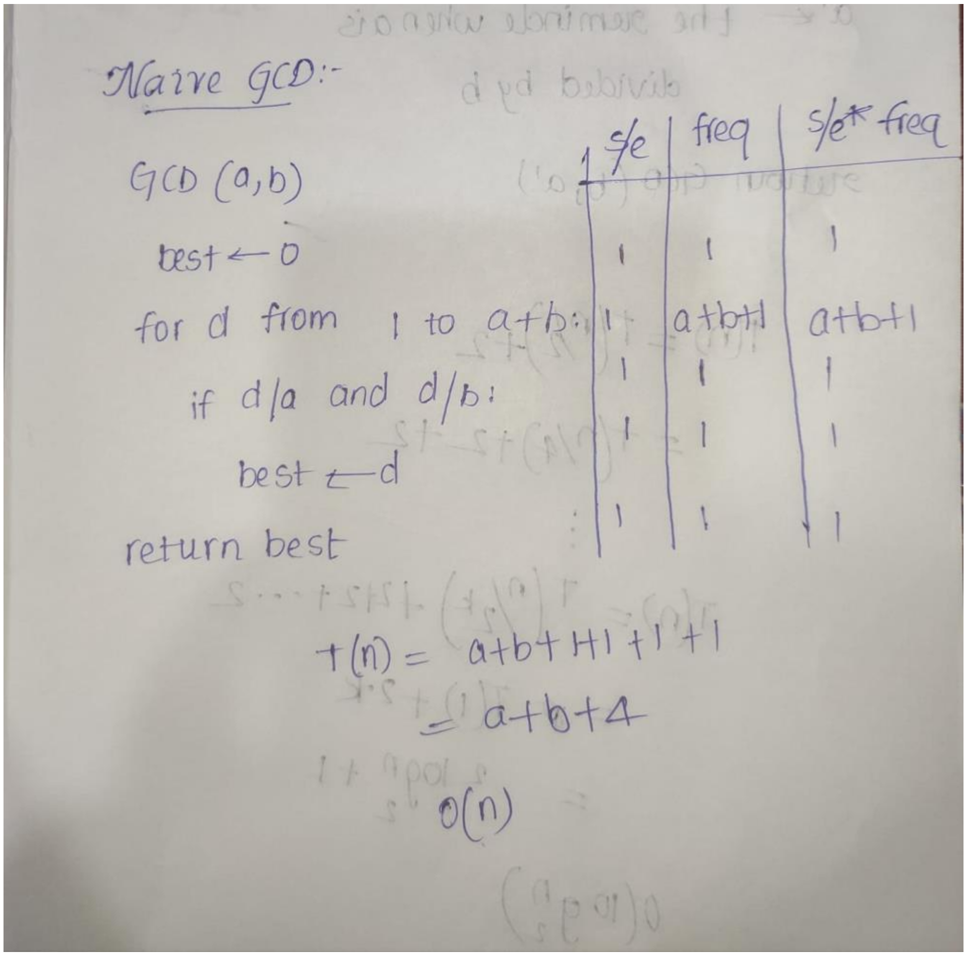
```
enter first number
100
enter second number
25
output:25


...Program finished with exit code 0
Press ENTER to exit console.
```



```

                                    Online Jav
                              Code, Compile, Run and
Write your code in this editor and pre

*************************************

import java.util.*;
public class Main{
public static void main(String args[])
```

```
enter first number
3918848
enter second number
1653264
output:61232


...Program finished with exit code 0
Press ENTER to exit console.
```

Asymptotic Analysis:



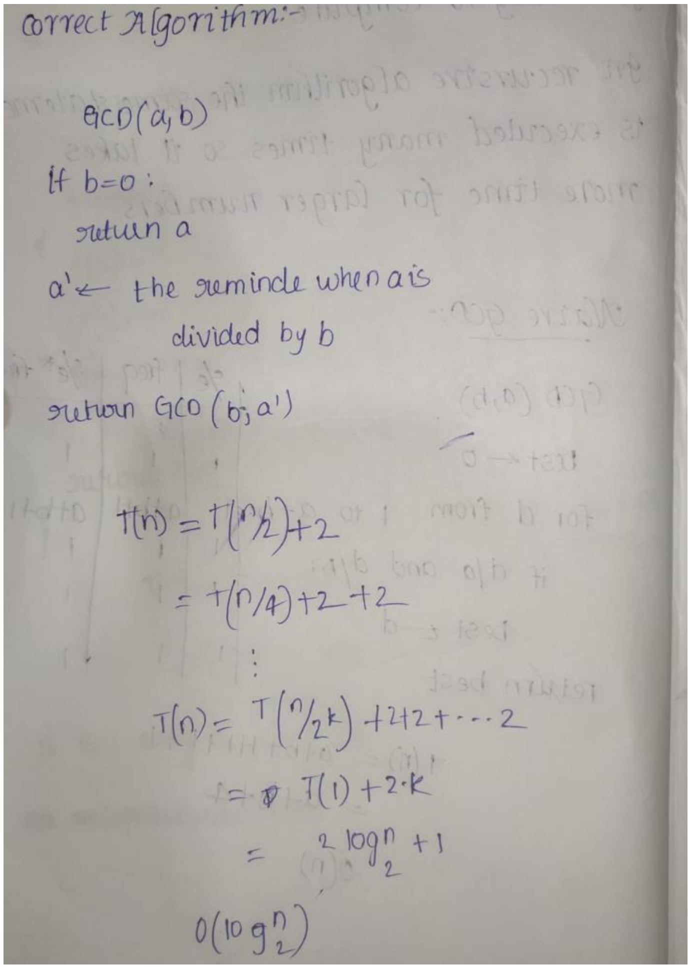
Naive GCD:-

	s/e	freq	s/e* freq
GCD (a,b)			
best ← 0	1	1	1
for d from 1 to a+b:	1	a+b+1	a+b+1
if d/a and d/b:	1	1	1
best ← d	1	1	1
return best	1	1	1

$t(n) = a+b+1+1+1+1$

$= a+b+4$

$O(n)$



correct Algorithm:-

```
GCD(a,b)
if b=0:
  return a
a' ← the reminde when a is
        divided by b
return GCD(b, a')
```

$T(n) = T(n/2)+2$

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

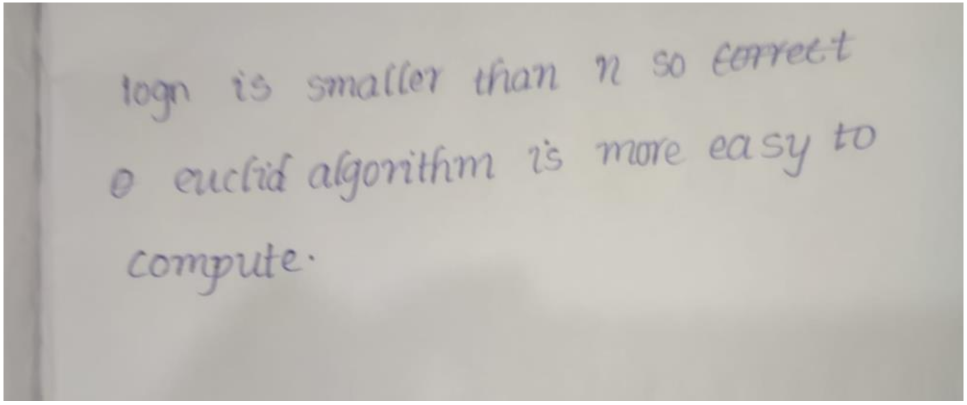
$\vdots$

$T(n) = T(n/2^k)+2+2+\cdots 2$

$= T(1)+2\cdot k$

$= 2\log_2^n + 1$

$O(\log_2^n)$



logn is smaller than n so correct euclid algorithm is more easy to compute.