

DESIGN AND ANALYSIS OF ALGORITHMS LAB3

Name:V.Prasanna Vyshnavi

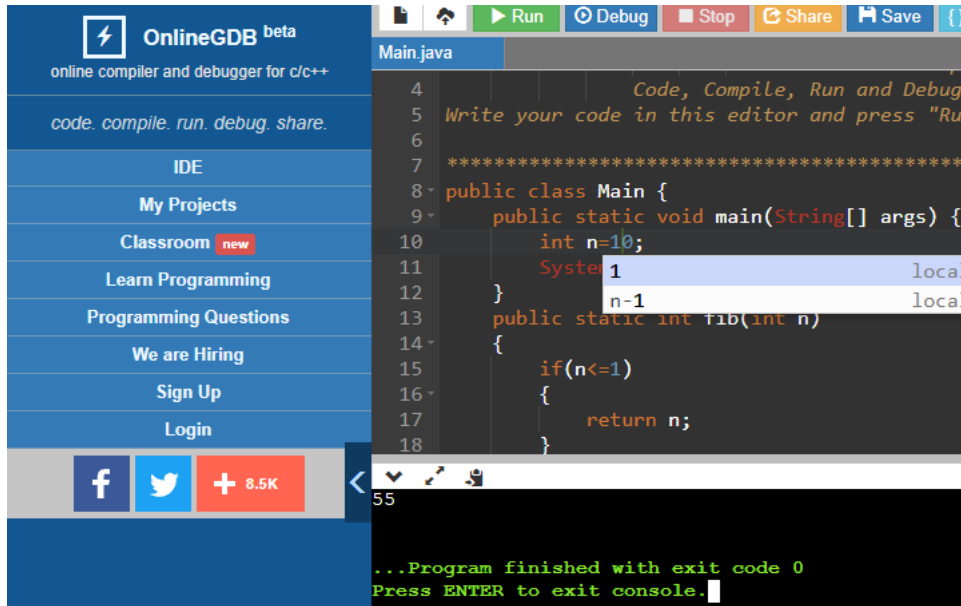
Regno:19BCE7661

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:



```
OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login

f 8.5K

Main.java
4
5 Code, Compile, Run and Debug
6 Write your code in this editor and press "Run" button to execute
7 *****
8 public class Main {
9     public static void main(String[] args) {
10         int n=10;
11         System.out.println(fib(n));
12     }
13     public static int fib(int n)
14     {
15         if(n<=1)
16         {
17             return n;
18         }
19     }
20 }

55

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



```
OnlineGDB beta
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login

f 8.5K

Main.java
3
4 Online Java Compiler.
5 Code, Compile, Run and Debug java program online
6 Write your code in this editor and press "Run" button to execute
7 *****
8
9 public class Main {
10     public static void main(String[] args) {
11         int n=300;
12         System.out.println(+ fib(n));
13     }
14     public static int fib(int n)
15     {
16         if(n<=1)
17         {
18             return n;
19         }
20     }
21 }

input
```

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:

The screenshot shows the OnlineGDB beta IDE interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, We are Hiring, Sign Up, and Login. Below these are social media icons for Facebook and Twitter, and a '+ 8.5K' button. The top of the IDE has a toolbar with buttons for Run, Debug, Stop, Share, and Save. The main editor area displays a Java file named 'Main.java' with the following code:

```
1 /*****  
2  
3 Online Java Compiler.  
4 Code, Compile, Run and Debug java pr  
5 Write your code in this editor and press "Run" butto  
6  
7 *****/  
8 import java.math.BigInteger;  
9 public class Main {  
10     public static void main(String[] args) {  
11         int n=10;  
12         System.out.println(fib(n));  
13     }  
14     public static BigInteger fib(int n)  
15     {
```

The console output at the bottom shows: "...Program finished with exit code 0" and "Press ENTER to exit console."

This screenshot shows the same OnlineGDB beta IDE interface as the first one, but with a different Java program. The code in 'Main.java' is:

```
1 /*****  
2  
3 Online Java Compiler.  
4 Code, Compile, Run and Debug java pr  
5 Write your code in this editor and press "Run" butto  
6  
7 *****/  
8 import java.math.BigInteger;  
9 public class Main {  
10     public static void main(String[] args) {  
11         int n=100;  
12         System.out.println(fib(n));  
13     }  
14     public static BigInteger fib(int n)  
15     {
```

The console output at the bottom shows: "...Program finished with exit code 0" and "Press ENTER to exit console." followed by a cursor.

Asymptotic Analysis:

Naive algorithm:-

if $n \leq 1$
return n

else

return $\text{fibRecurs}(n-1) + \text{fibRecurs}(n-2)$

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

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

$$x^n = x^{n-1} + x^{n-2} = (n)T$$

$$x^2 = x + 1 = (001)T$$

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

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

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

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

Effective Algorithm:-

fiblist(n)

{
 f[0] ← 0

 f[1] ← 1

 for i from 2 to n:

 f[i] ← f[i-1] + f[i-2]

 return f[n]

}

%	freq	%*freq
1	1	1
1	1	1
1	n	n
1	n-1	n-1
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:

```
1  /*****
2
3
4      Online Java Compiler.
5      Code, Compile, Run and Debug java
6      Write your code in this editor and press "Run" button
7
8  *****/
9  import java.util.*;
10 public class Main{
11     public static void main(String args[])
12     {
13         Scanner sc = new Scanner(System.in);
14         System.out.println("enter first number");
15         int a = sc.nextInt();
16         System.out.println("enter second number");
17         int b = sc.nextInt();
18         System.out.println("output:" + GCD(a,b));
19     }
20     static int GCD(int a,int b)
21     {
22         int best=0;
23         for(int d=1;d<=a+b;d++)
24         {
25             if(a%d==0 && b%d==0)
26                 best=d;
27         }
28         return best;
29     }
30 }
```

enter first number
75
enter second number
25
output:25

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

```
15     int a=sc.nextInt();
16     System.out.println("enter second number");
17     int b=sc.nextInt();
18     System.out.println("output:" + GCD(a,b));
19 }
20 static int GCD(int a,int b)
21 {
22     int best=0;
23     for(int d=1;d<=a+b;d++)
24     {
25         if(a%d==0 && b%d==0)
26             best=d;
27     }
28     return best;
29 }
```

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:

 **OnlineGDB** beta

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new




Learn Programming








Programming Questions

We are Hiring

Sign Up




Login

   8.5K

   Run  Debug  Stop  Share 

Main.java

```
1  /*****
2
3
4      Online Java
5      Code, Compile, Run and
6      Write your code in this editor and pre
7      *****/
8
9  import java.util.*;
10 public class Main{
11 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.

code. compile. run. debug. share.

IDE

My Projects

Classroom new




Learn Programming

Programming Questions




We are Hiring

Sign Up

Login

   8.5K

```
2
3
4      Online Java
5      Code, Compile, Run and
6      Write your code in this editor and pre
7      *****/
8
9  import java.util.*;
10 public class Main{
11 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:-

$\text{gcd}(a, b)$

$\text{best} \leftarrow 0$

for d from 1 to $a+b$:

if $d|a$ and $d|b$:

$\text{best} \leftarrow d$

return best

s/e	freq	$s/e * \text{freq}$
1	1	1
$a+b+1$	1	$a+b+1$
1	1	1
1	1	1
1	1	1

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

$$= a+b+4$$

$$O(n)$$

correct algorithm:

$\text{gcd}(a, b)$

if $b=0$:

return a

$a' \leftarrow$ the remainder when a is
divided by b

return $\text{gcd}(b, a')$

$$T(n) = T\left(\frac{n}{2}\right) + 2$$

$$= T\left(\frac{n}{4}\right) + 2 + 2$$

\vdots

$$T(n) = T\left(\frac{n}{2^k}\right) + 2 + 2 + \dots + 2$$

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

$$= 2 \log_2 n + 1$$

$$O(\log_2 n)$$

$\log n$ is smaller than n so correct
Euclid algorithm is more easy to
compute.