```
S.o.pln(info);
}
}
```

21/1/25

```
* public class student {
    static student1 updateInfo(student s) {
        s.name = "bhaggi";
        s.id = 992;
        s.cgp = 9.8f;
    return s;
}
    p.s.v.m (string[] args){
        student1 s1 = new student1();
        S.o.pln("original Information:");
        S.o.pln("s1.sname");
        S.o.pln("s1.id");
        S.o.pln("s1.cgp");
        updateInfo(s1);
        S.o.pln("updated Information:");
        S.o.pln("s1.sname");
        S.o.pln("s1.id");
        S.o.pln("s1.cgp");
    }
}

class student1{
    int id = 992;
    string name = "bhaggi";
    float cgp = 9.8f;
}
```

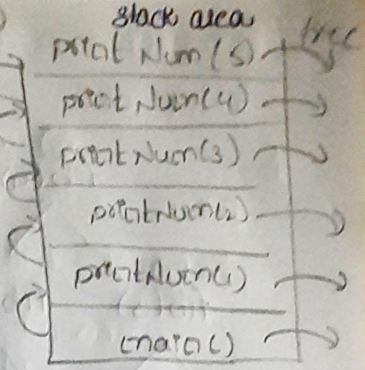
* pythontutor.com

Divide complex into small task

Recursion (code reusability)

A function call itself

```
Package recursion;
Public class RecursionExample {
    static void printNum(int n){
        if(n==5){
            S.o.pln(n);   } Base condition
            return;
        }
        S.o.pln(n);
        printNum(n+1);
    }
    static void printNum5(){
        S.o.pln(5);
    }
    public static void main(String [] args){
        printNum(1);
    }
}
```

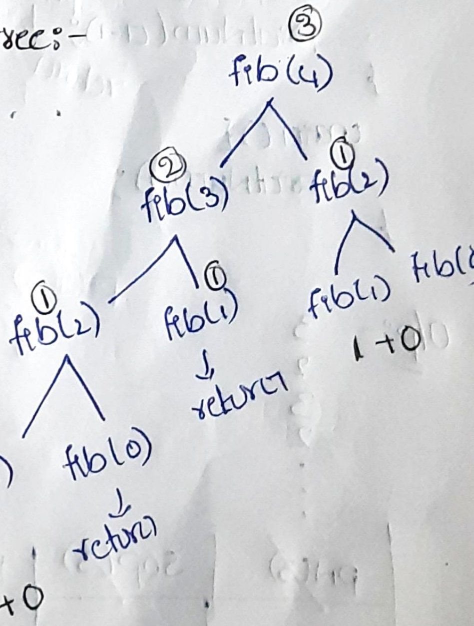


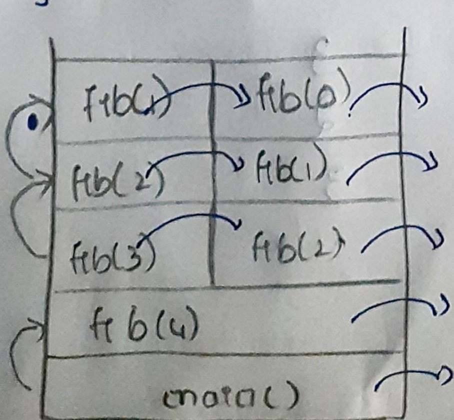
O/p:- 1
2
3
4
5

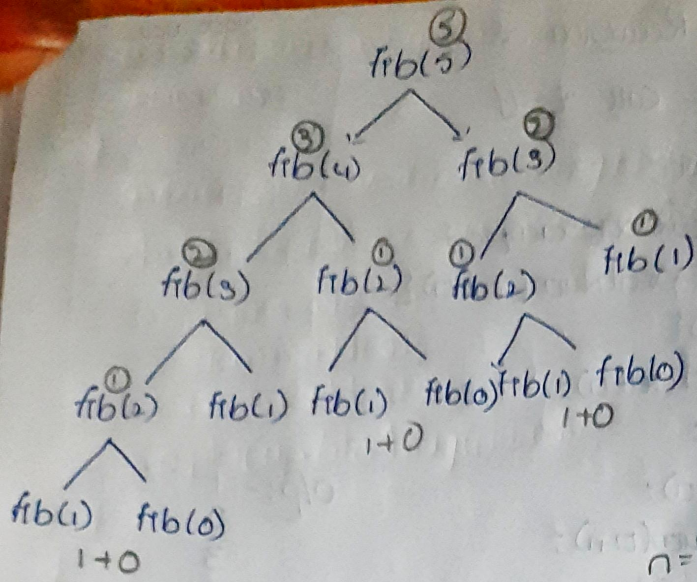
* fibonacci with recursive tree:-

0, 1, 1, 2, 3, 5, 8, ...

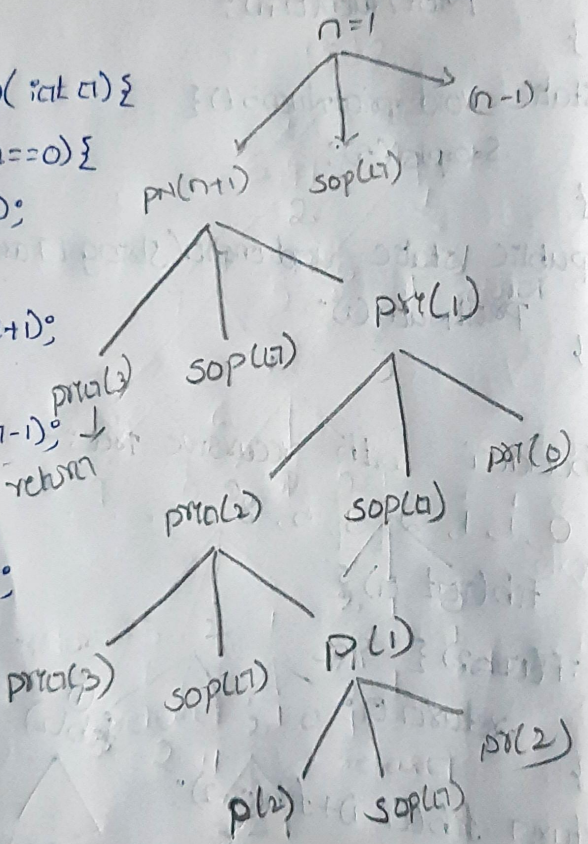
```
int fib(int n){
    if(n<2){
        return n;
    }
    return fib(n-1)+fib(n-2);
}
```








```
* void printNum(int n){
    if(n==3 || n==0){
        S.O.P(n);
        return;
    }
    printNum(n+1);
    S.O.P(n);
    printNum(n-1);
}
main(){
    printNum(1);
}
```



O/P:- 3
2
3
2
⋮

~~PN(3)~~	~~SOP(3)~~	PN(2)
~~PN(2)~~	~~SOP(2)~~	PN(0)
~~PN(3)~~	~~SOP(3)~~	PN(1)
printNum(2)	SOP(2)	PN(0)
printNum(1)		
main()		

3
2
3
order

* N = 14235

T.C:- $\log_{10} n$

5 + sum(1423)

sum(N) = N%10 + sum(N/10) 3 + sum(142)

2 + sum(14)

4 + sum(1)

1 + 0

```
* Public class RecursiveExample {
    static int digitSum(int n){
        if(n==0){
            return 0;
        }
        return n%10+digitSum(n/10);
    }
    static void printNum(int n){
        if(n==5){
            S.o.println(n);
            return;
        }
        S.o.println(n);
        printNum(n+1);
    }
    static void printNum5 (){
        S.o.println(5);
    }
    Public static void main(String [] args){
        printNum(1);
    }
}
```

```
* package recursion;
  import java.util.ArrayList;
  public class Student{
    static ArrayList <Integer> updateArray(ArrayList
        <Integer > arr){

      int sum=0;
      for(int i=0; i<arr.size(); i++)
        sum+=arr.get(i);
      for(int i=0; i<arr.size(); i++)
```

```
        arr.set(i, sum - arr.get(i));
    return arr;
}

Public static void main(String [] args) {
ArrayList < Integer > nums = new ArrayList();
        nums.add(1);
        nums.add(2);
        nums.add(3);
    S.O.println(updateArray(nums));
}
}
```