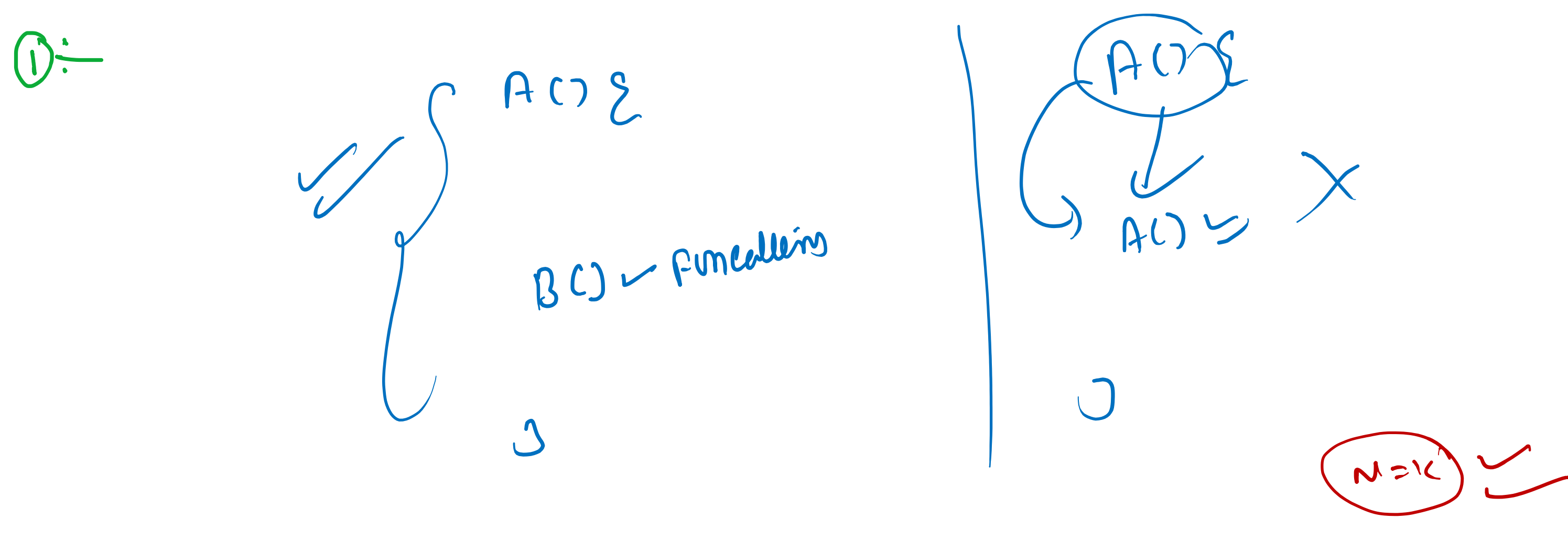


① Function Calling

Base calling



PMI → Mathematical proof

Base Case

① Smaller → Input

② N=k ✓

③ N=k+1 ✓

$$\sum_{i=1}^N i = 1+2+3+\dots+N = \frac{N(N+1)}{2}$$

$$\sum_{i=1}^{k+1} i = 1+2+3+\dots+k+(k+1) = \frac{(k+1)(k+2)}{2}$$

$$\frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2}$$

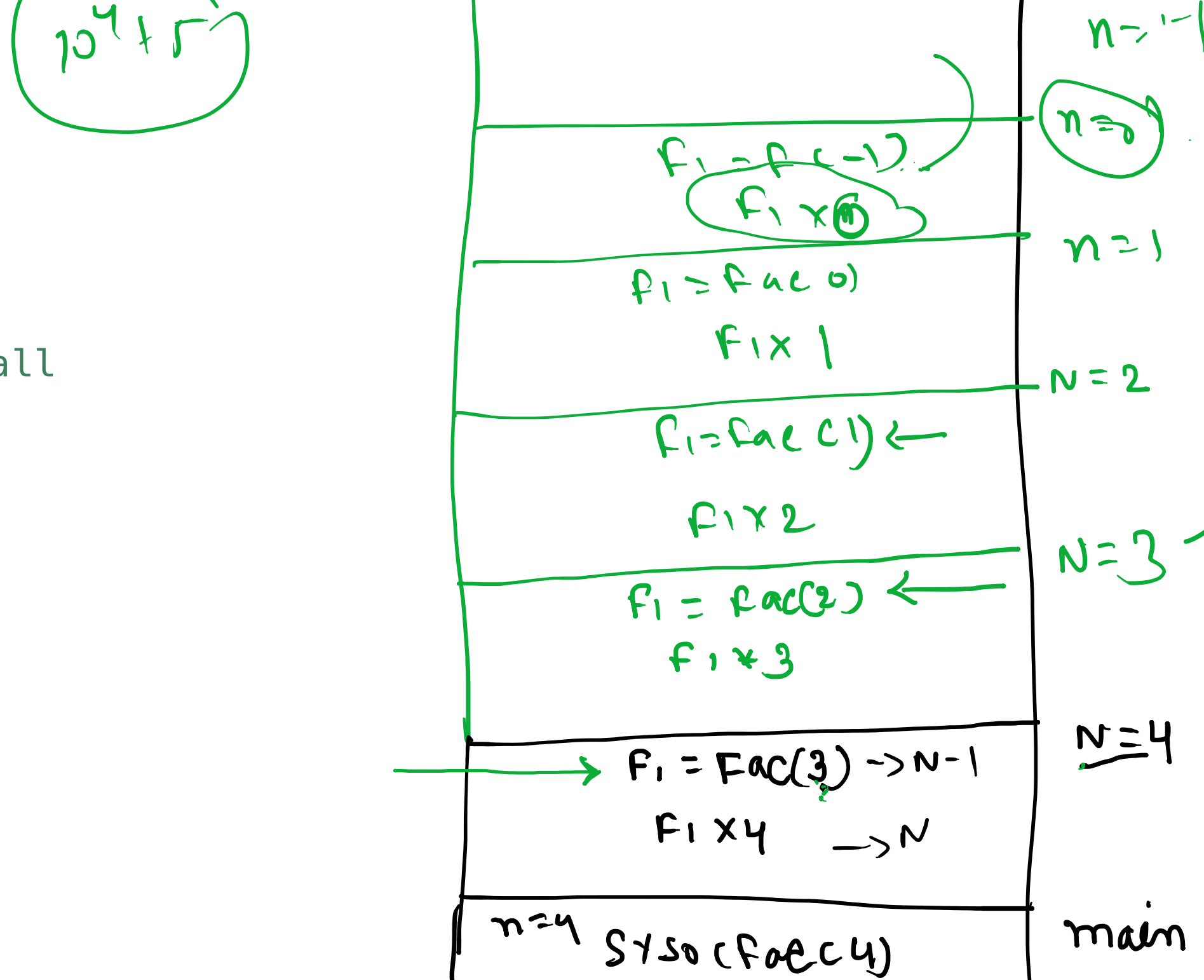
$$(k+1) \left\{ \frac{k}{2} + 1 \right\} = \frac{(k+1)(k+2)}{2}$$

$$(k+1) \left\{ \frac{k+2}{2} \right\} = \frac{(k+1)(k+2)}{2}$$

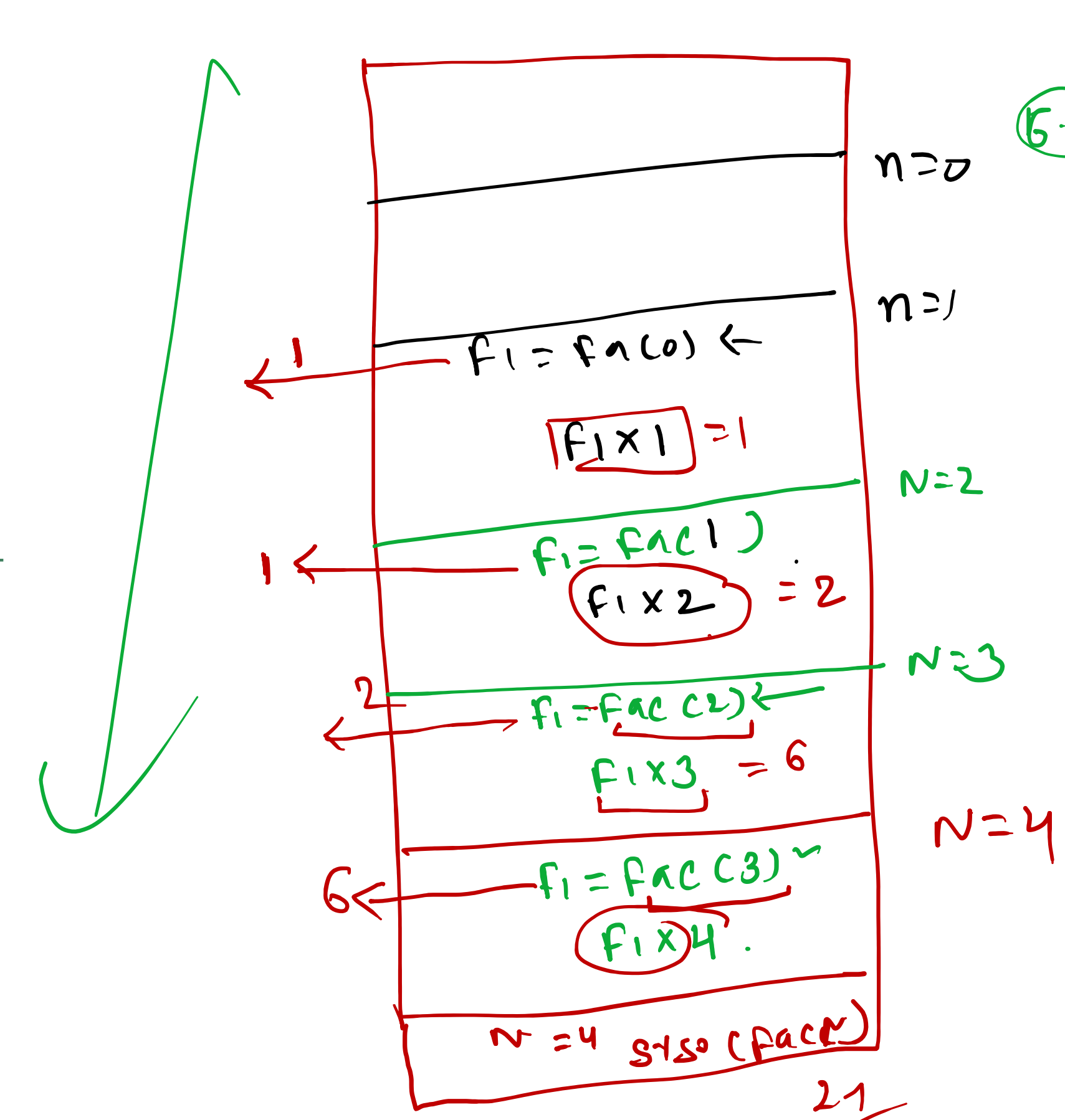
$$N! = N \times (N-1)!$$

$$5! = 5 \times 4! = 4 \times 3! = 3 \times 2! = 2 \times 1! = 1 \times 0! = 1$$

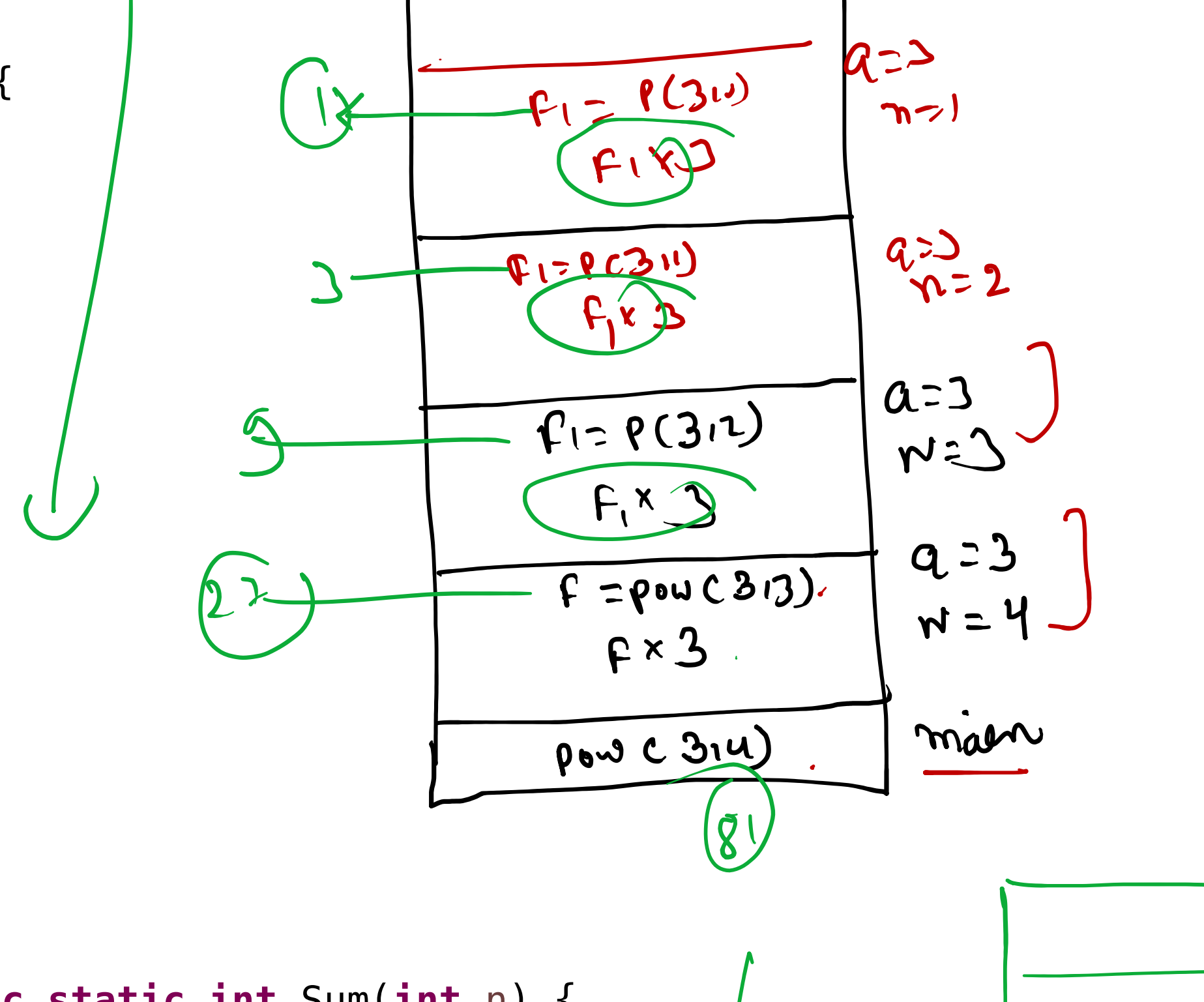
```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 4;  
    System.out.println(Fac(n));  
}  
  
public static int Fac(int n) {  
    int f1 = Fac(n-1); // recursive call  
    return f1*n;  
}
```



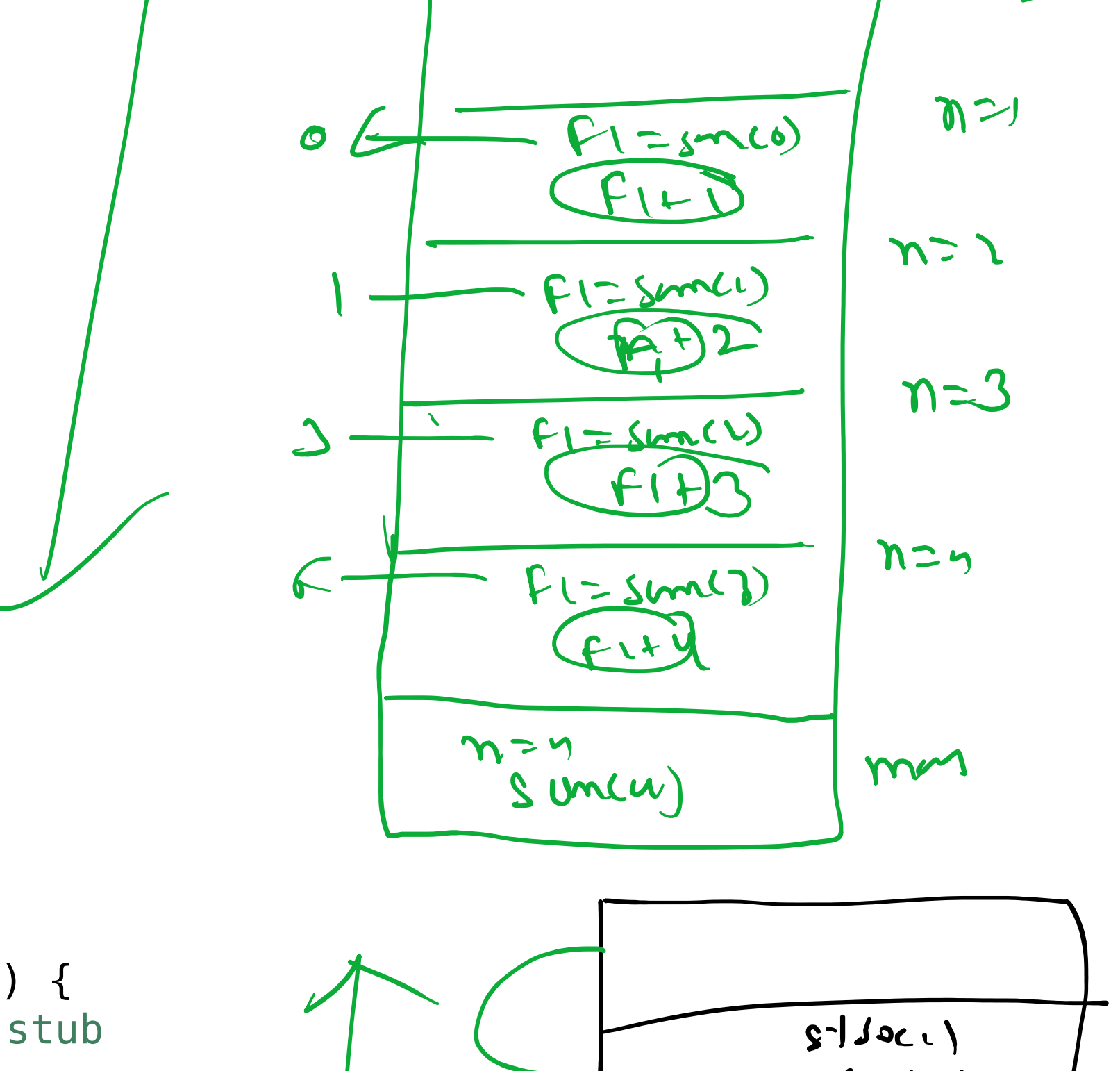
```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 4;  
    System.out.println(Fac(n));  
}  
  
public static int Fac(int n) {  
    if (n == 0) {  
        return 1;  
    }  
    int f1 = Fac(n-1); // recursive call  
    return f1 * n;  
}
```



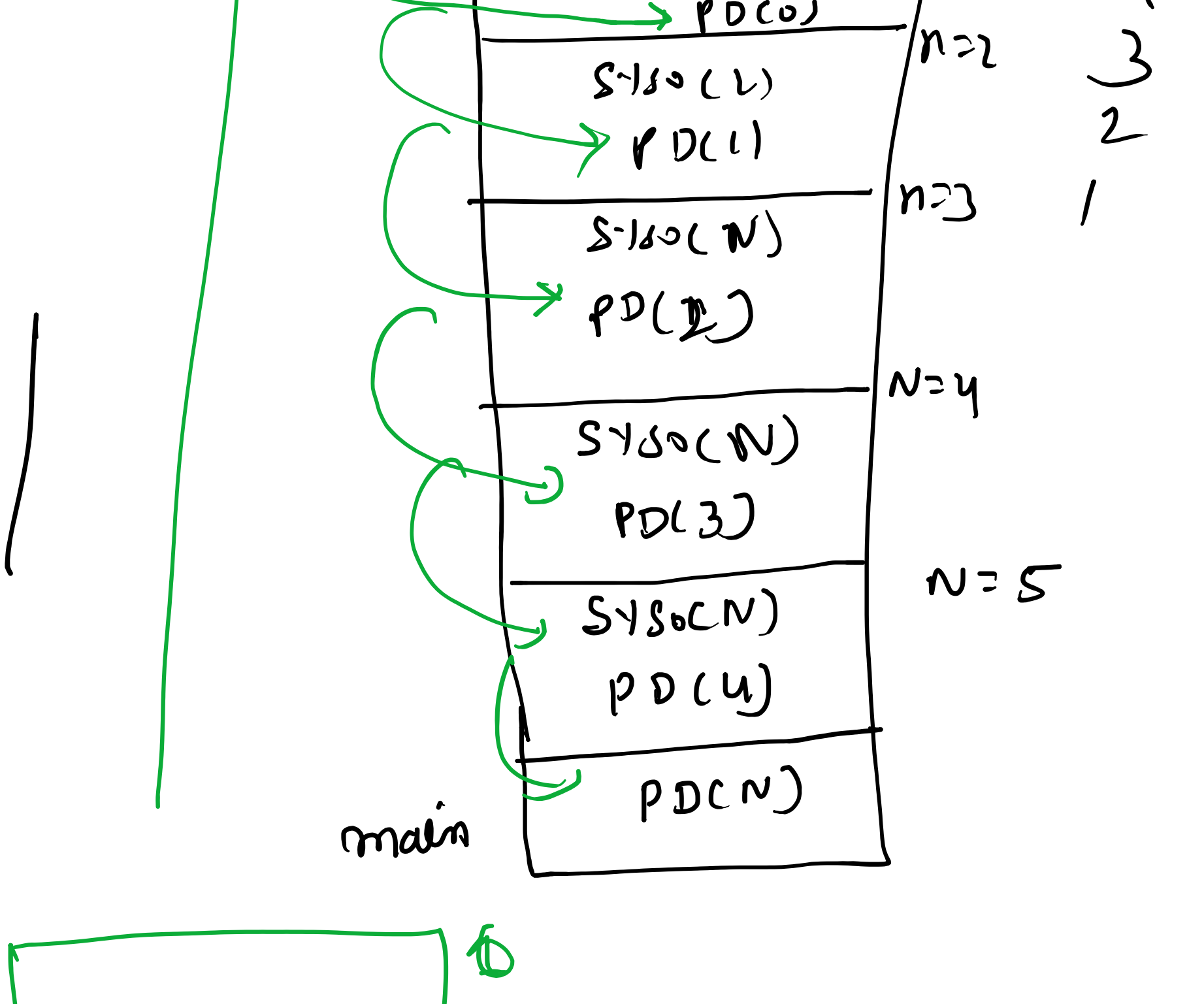
```
public static int pow(int a, int n) {  
    if (n == 0) {  
        return 1;  
    }  
    int f = pow(a, n-1);  
    return f*a;  
}
```



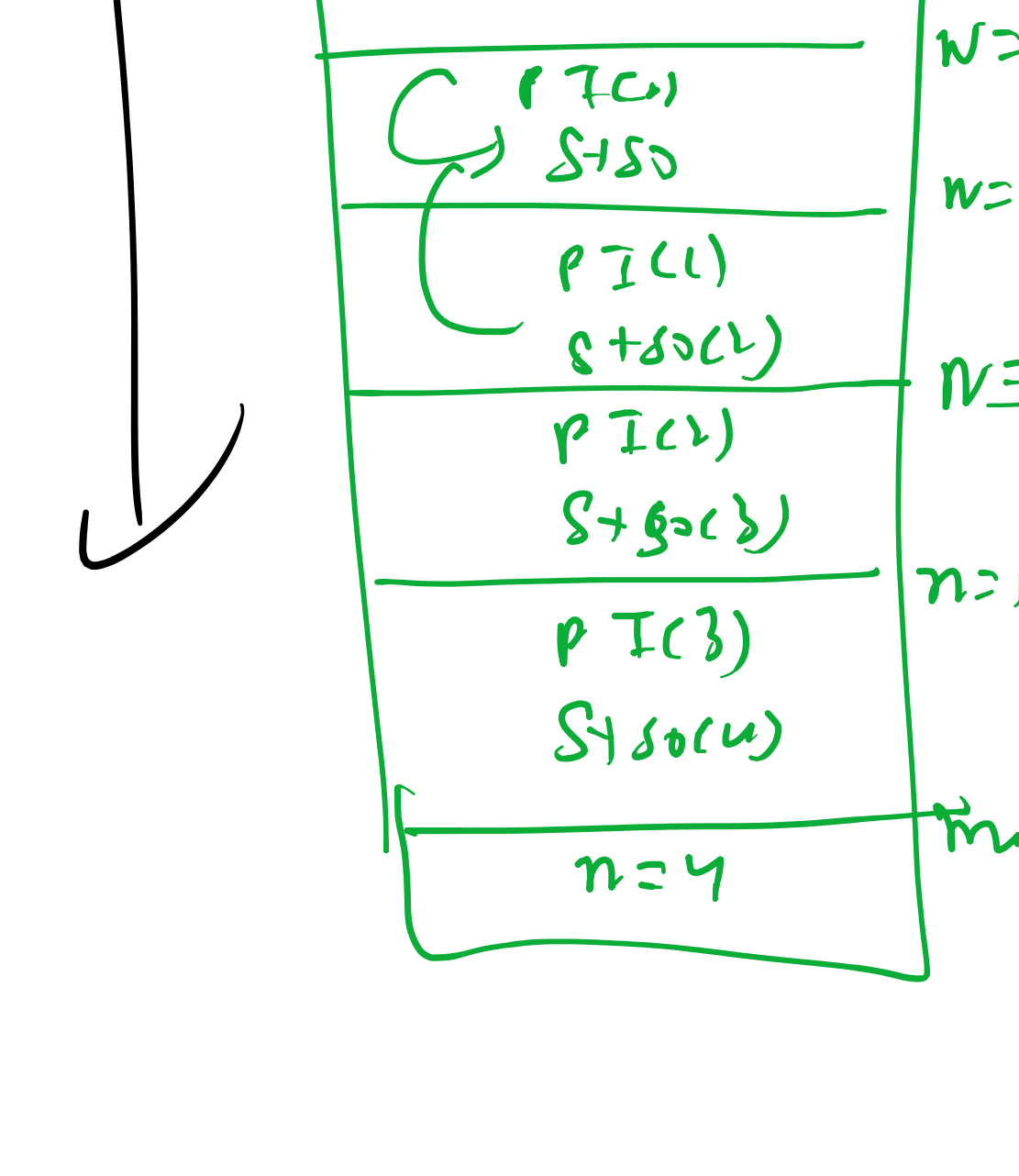
```
public static int Sum(int n) {  
    if (n == 0) {  
        return 0;  
    }  
    int f = Sum(n-1);  
    return f + n;  
}
```



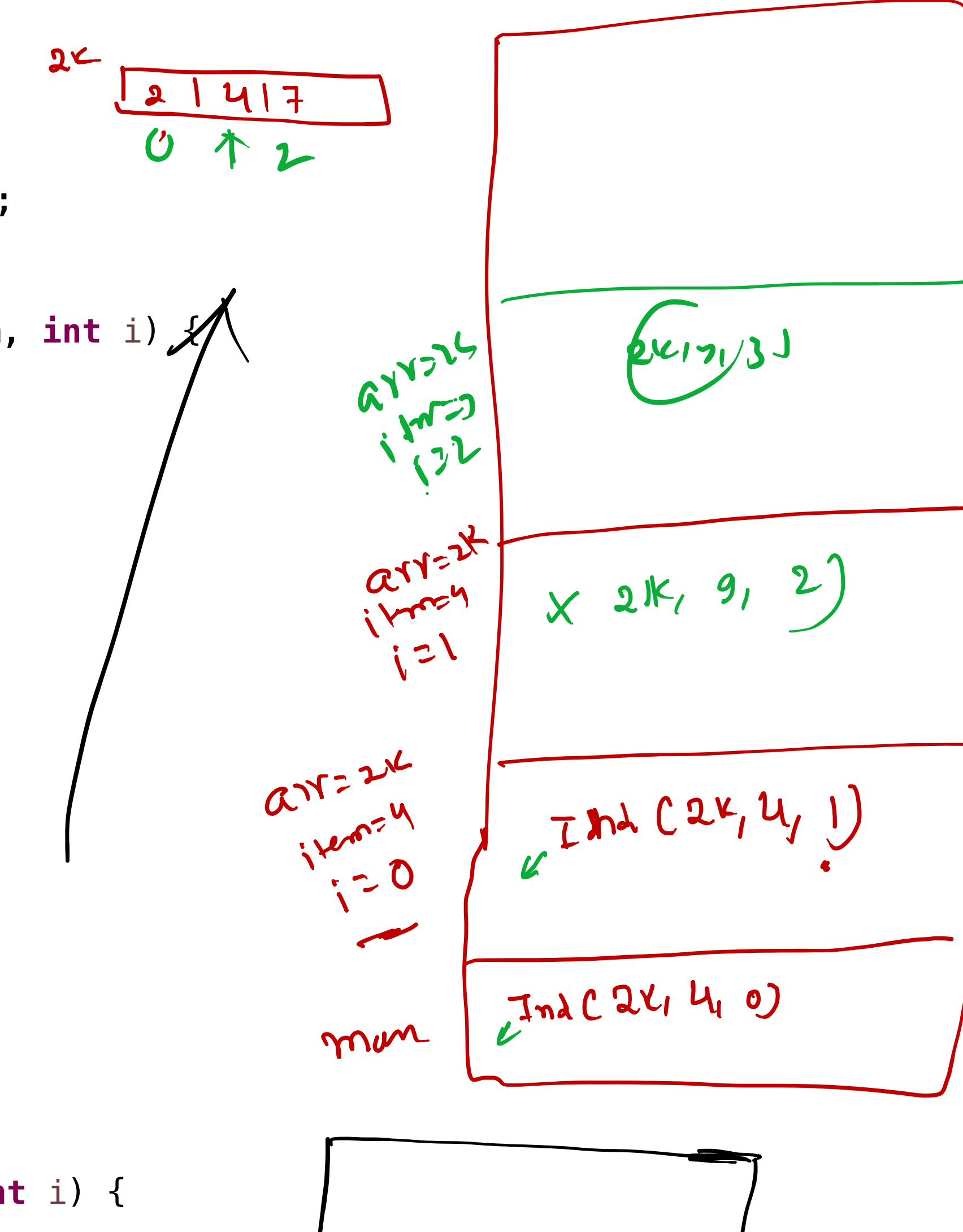
```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 5;  
    PD(n);  
}  
  
public static void PD(int n) {  
    if (n == 0) {  
        return;  
    }  
    System.out.println(n);  
    PD(n-1);  
}
```



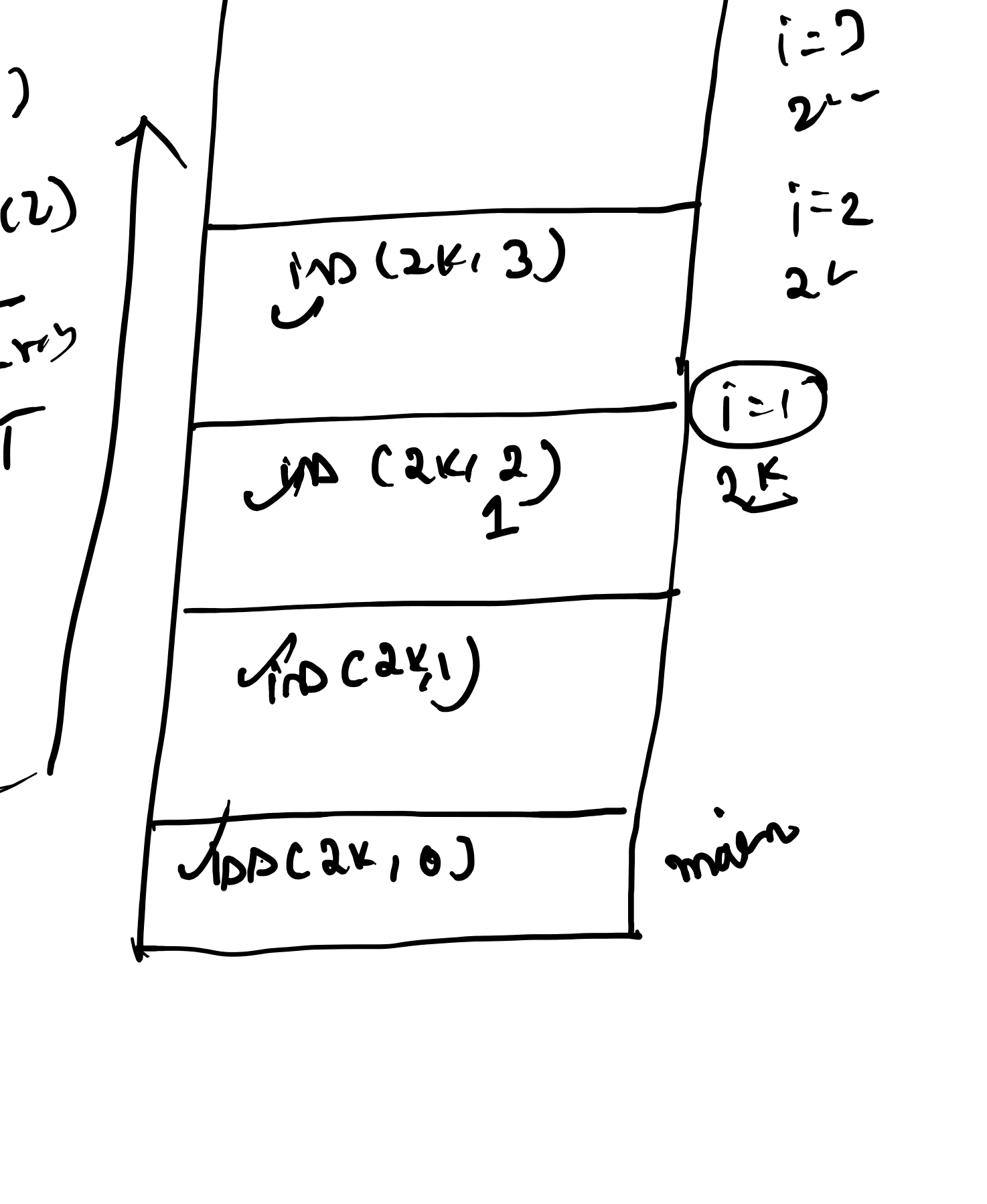
```
public static void PI(int n) {  
    if (n == 0) {  
        return;  
    }  
    PI(n-1);  
    System.out.println(n);  
}
```



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int[] arr = { 2, 3, 4, 6, 5, 4, 7, 4 };  
    int item = 4;  
    System.out.println(index(arr, item, 0));  
}  
  
public static int index(int[] arr, int item, int i) {  
    if (arr[i] == item) {  
        return i;  
    }  
    return index(arr, item, i+1);  
}
```

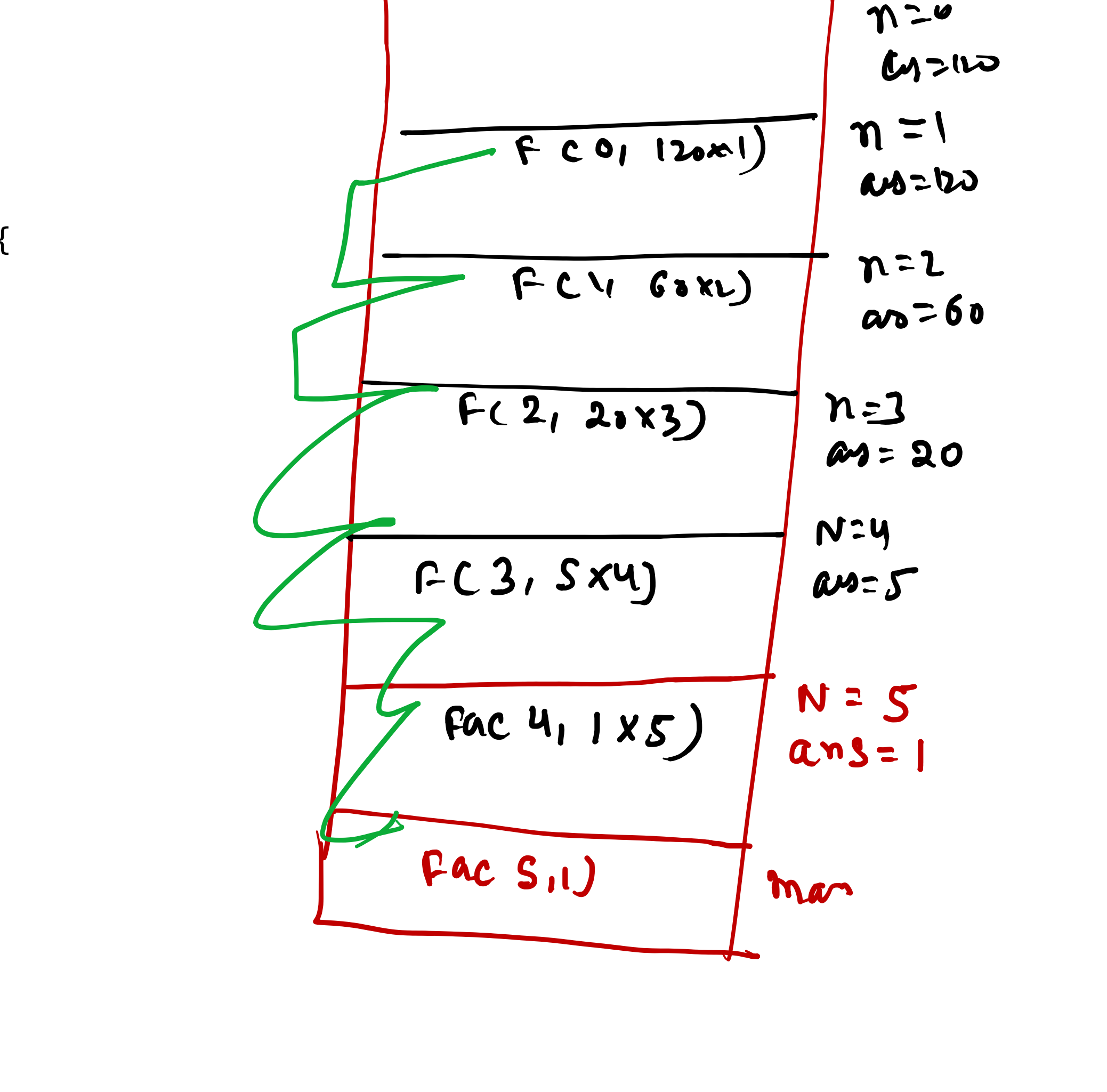


```
public static boolean issorted(int[] arr, int i) {  
    if (arr.length - 1 == i) {  
        return true;  
    }  
    if (arr[i] > arr[i+1]) {  
        return false;  
    }  
    return issorted(arr, i+1);  
}
```



tail  
rec

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 5;  
    System.out.println(Fac(n, 1));  
}  
  
public static int Fac(int n, int ans) {  
    if (n == 0) {  
        return ans;  
    }  
    return Fac(n-1, ans * n);  
}
```



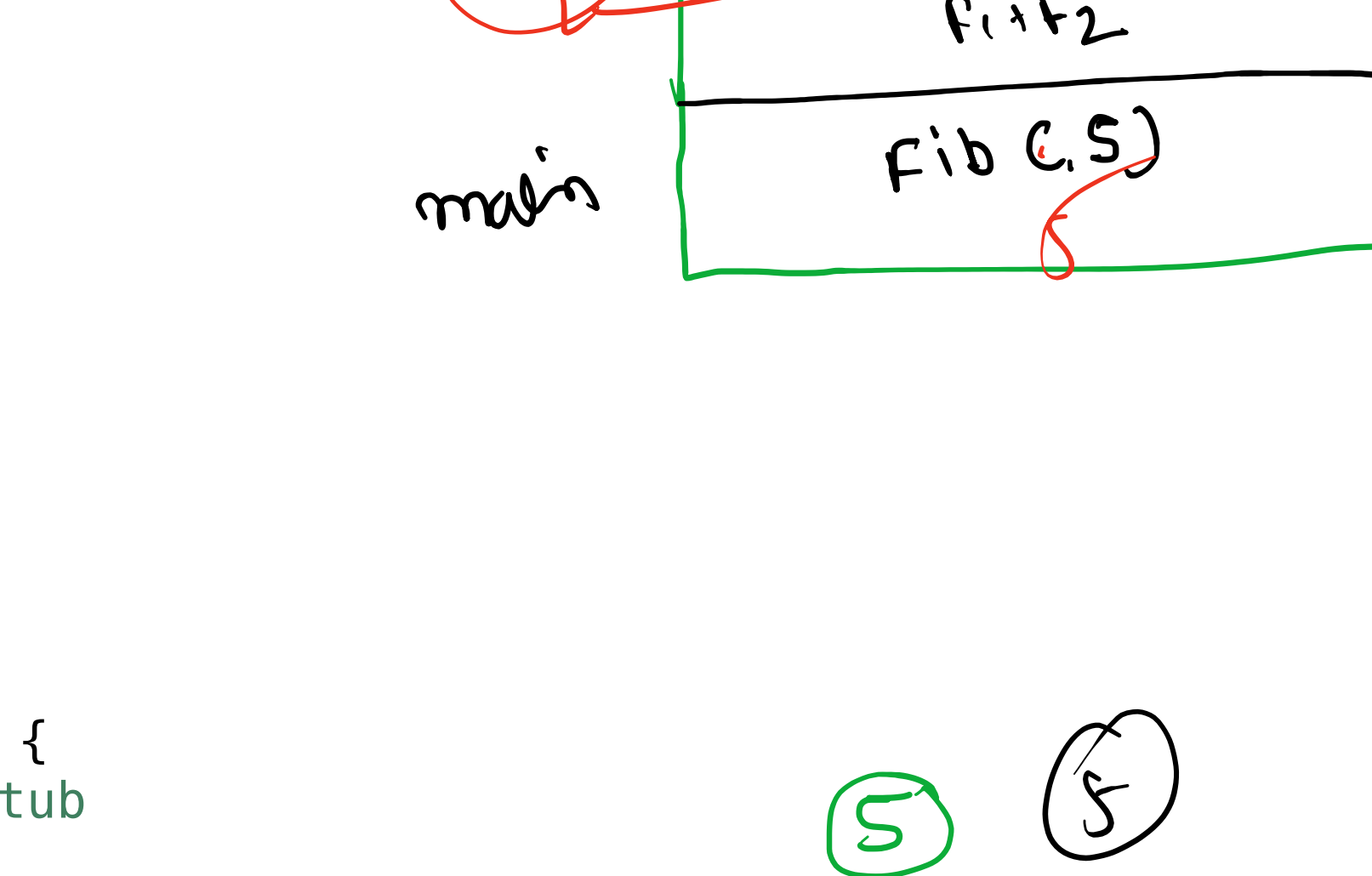
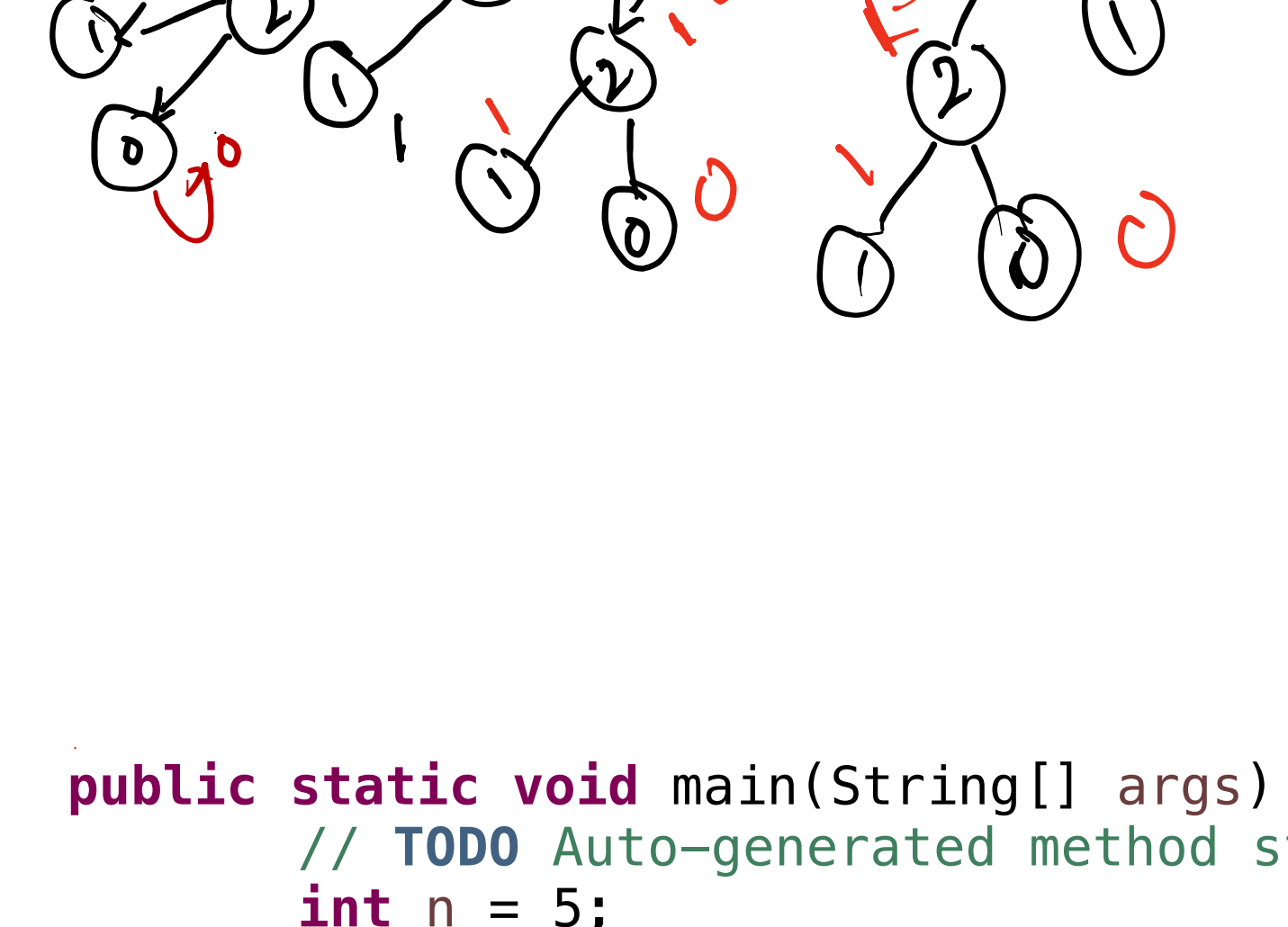
0 1 1 2 3 5 8 13 21 34 55 89 144 233 ---

0 1 2 3 4 5 6

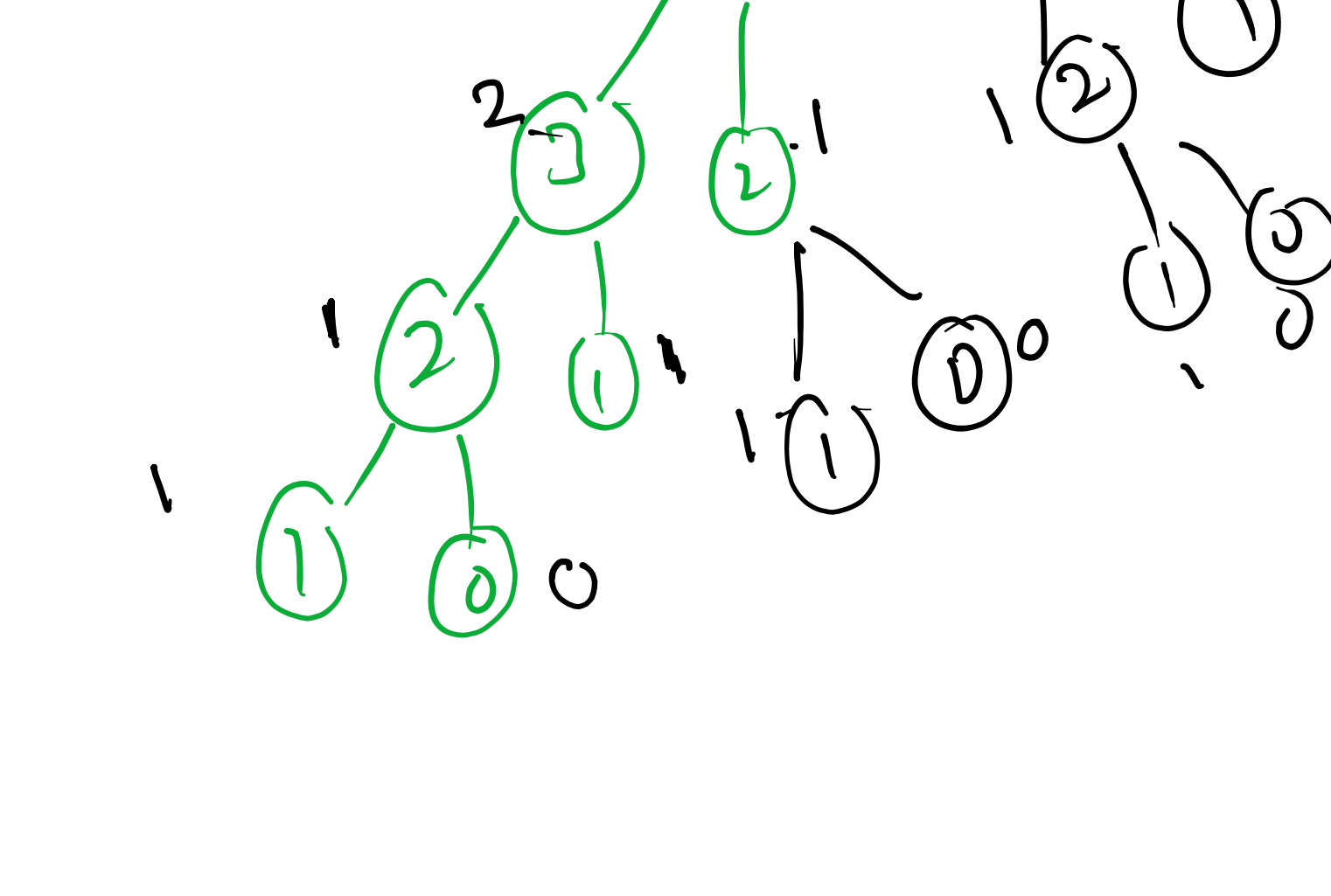
$$fib(n) = fib(n-1) + fib(n-2)$$

$$Fib(n-1) \rightarrow F_1$$
  
$$Fib(n-2) \rightarrow F_2$$
  
$$F_1 + F_2$$

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 5;  
    System.out.println(Fib(n));  
}  
  
public static int Fib(int n) {  
    if (n == 0 || n == 1) {  
        return n;  
    }  
    int f1 = Fib(n-1);  
    int f2 = Fib(n-2);  
    return f1 + f2;  
}
```



```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    int n = 5;  
    System.out.println(Fib(n));  
}  
  
public static int Fib(int n) {  
    if (n == 0 || n == 1) {  
        return n;  
    }  
    int f1 = Fib(n-1);  
    int f2 = Fib(n-2);  
    return f1 + f2;  
}
```



①  $\lambda = 3247$

②  $\lambda = 3247$