


Recursion ↳ when a function calls itself

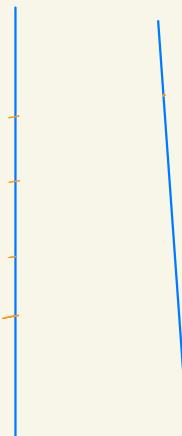
- 1) steps
- ⇒ solve it for the smallest problem.
 - ⇒ Assume that it will work properly for smaller problem (faith)
 - ⇒ Solve it for a smaller solution (your work)

Ques Take an input x , print numbers from x to 1.

⇒ public static void fun(int x) {
 if ($x == 1$) {
 base case
 System.out.println(x);
 return;
 }

step ① ↓
 {
 if ($x == 1$) {
 base case
 System.out.println(x);
 return;
 }
 }
 ↓
 Sys.out(x);

step ② ↓
 fun(x-1);
 ↓
 {



$x-1$

Expectation

it will print numbers from x to 1
properly

work

✓ Faith

it will print numbers from $(x-1)$ to 1

$x=5$

$\frac{5}{4}$
3
2
1

faith $(x-1)$

{ 4
3
2
1

Ques Print numbers from 1 to x

work

Expectation (x) $x=5$ fact

It will print numbers from 1 to x

property

I to well print numbers from 1 to $x-1$ property.

{
1
2
3
4
5

psv print (int x) {

- 1) print ($x-1$);
- 2) System.out.println (x);

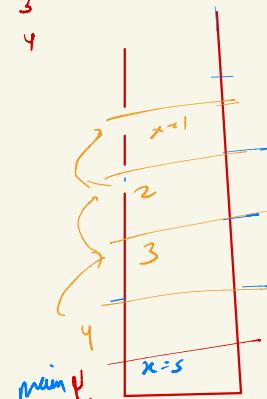
if ($x == 1$) {

System.out.println (x);

return;

}

{
1
2
3
4



1
2
3
4
5

psv print (int x) {

if ($x == 1$) {

System.out.println (x);

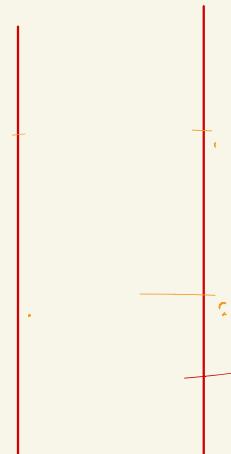
return;

}

print ($x-1$);

System.out.println (x);

}



1
2
3
4

Ques Print numbers from n to 1 using recursion.

print Decreasing

Expectation $\frac{1}{n} \sum_{i=1}^n i$
It will print numbers from n to 1

5
4
3
2
1

psv $\text{pd}(\text{int } n)$ of

1) $\text{Sprint}(n)$
2) $\text{pd}(n-1)$

3)
 $y(n=1)$ of
 $\text{Sprint}(n)$
return;

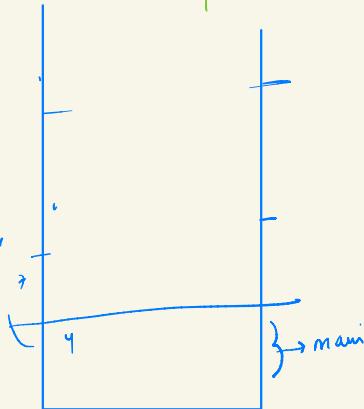
Does $n=5$
first print
decreasing them
Inversely

5
4
3
2
1
2
3
4
5

$n=4$)

Fact
It will print numbers from $(n-1)$ to 1

4
3
2
1



4
3
2
1

4
3
2
1
2
3
4

expectation

(n=5)

It will print decreasing
from $n \rightarrow 1$, then $1 \rightarrow n$

faith

(n=4)

It will print $(n-1)$ to 1
and then 1 to $(n-1)$

5
4
3
2
1
2
3
4
5

```
public static void printDecreasingIncreasing(int x){
    if(x==1){
        System.out.println(x);
        return;
    }
    System.out.println(x);
    printDecreasingIncreasing(x-1);
    System.out.println(x);
```

4
3
2
1
2
3

$\{$
 $\text{if } (x==1)$
 $\text{System.out.println(x)}$
 return



3
2
1
2
3

Ques Write a function to find factorial $n!$

expectation

(n=5)

It will return 120
 $n!$

$5 \times 4 \times 3 \times 2 \times 1$

public static int fac (int n) {
 if ($n == 0$) return 1;

1) int f = fac (n-1);
2) int ans = n * f;
return ans;

faith

It will return
 $(n-1)!$

4
 \downarrow
 $4 \times 3 \times 2 \times 1$
 \downarrow
24

Ques Write function to calculate a raise to power b

Expectation $a=2$
 $b=4$

It will calculate and
return a^b

$$\begin{matrix} 2^4 \\ \downarrow \\ 2 \times 2 \times 2 \times 2 \end{matrix}$$

ps int power(int a, int b){
1) if ($b==0$) return 1;
2) int f = power(a, b-1);
3) int ans = a * f
return ans;

$O(b)$

}

Expectation $a=2, b=4$

It will calculate a^b .

$$\overbrace{2 \times 2 \times 2 \times 2}$$

$$ans = f \times f$$

ps int power - better (a, b){
1) if ($b==0$) return 1;
2) int f = pb(a, b/2);
3) int ans = f * f;
if ($b \cdot 1.2 == 1$) {
 ans = a * ans;
}
return ans;

$O(\log b)$

}

$$f=9$$

$$\begin{aligned} ans &= 9 \times 9 \times 81 \\ ans &= 3 \times 81 \Rightarrow 243 \end{aligned}$$

main {
 a=2, b=1
}

$\boxed{a=3, b=5}$ $\boxed{243}$

$$b \rightarrow \frac{b}{2} \rightarrow \frac{b}{4} \rightarrow \frac{b}{8} \dots \textcircled{1}$$

$$\frac{b}{2^1}, \frac{b}{2^2}, \frac{b}{2^3} \dots \textcircled{\frac{b}{2^x}}$$

$$\frac{b}{2^x} = 1 \Rightarrow b = 2^x$$

$$\Rightarrow \frac{\log b}{\log 2} = x \Rightarrow x = \underline{\underline{\log_2 b}}$$

Ques Print array using recursion

$$\text{arr} = [10, 29, 11, 5, 43]$$

Expectation
Print elements from index 0

to $(n-1)$:

10

29

11

5

43

Faith

Print elements from index
1 to $n-1$

$\downarrow (\text{idn} = \text{arr.length}) \text{return}$

psv print_array (int arr, int idn) {

- 1) System.out.println (arr[idn]);
- 2) print_array (arr, idn+1);

)

Recursion on the way up

10

29

11

5

43

main {
 arr, 0
}

$0 \rightarrow n-1$ reverse

43
5
11
29
10

10, 29, 11, 5, 43

for (arr, idn) {
 for (arr, idn+1);
 System.out.println(idn);
}

$i \rightarrow n-1$ reverse

43
5
11
- 29

43
5
11
29
10

