

Lec-15
03/03/25 9:49 AM

Recursion

functioning it self

PMI

Mathematical Proof

#1

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

base case
N=1: $i = 1 = \frac{1 \times (1+1)}{2}$
N=2: $i = 1+2 = \frac{2 \times (2+1)}{2}$
N=k: $i = 1+2+3+\dots+k = \frac{k \times (k+1)}{2}$ (S.P.)

#2

$$i = 1 + 2 + 3 + \dots + k + (k+1)$$

base case
N=1: $i = 1 = \frac{1 \times (1+1)}{2}$
N=2: $i = 1+2 = \frac{2 \times (2+1)}{2}$
N=k: $i = 1+2+3+\dots+k = \frac{k \times (k+1)}{2}$
N=k+1: $i = 1+2+3+\dots+k+(k+1) = \frac{(k+1) \times (k+2)}{2}$

fun calling inside

AOL
AOL
AOL

AOZ
BOZ

Recursion

3 steps

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

Self work
2nd step
PMI

Recursion

3 steps

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

Self work
2nd step
PMI

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

public static int fac(int n) {
// base case;
if (n == 0) {
return 1;
}
int fn = fac(n-1);
return fn * n;
}

public static void main(String[] args) {
// TODO Auto-generated method stub
int a = 3;
int n = 5;
System.out.println(pow(a, n));
}

public static int pow(int a, int n) {
// base case;
if (n == 0) {
return 1;
}
int ans = pow(a, n-1);
return ans * a;
}

public static void main(String[] args) {
// TODO Auto-generated method stub
int n = 5;
PD(n);
}

public static void PD(int n) {
// base case;
if (n == 0) {
return;
}
System.out.println(n);
PD(n-1);
}

public static void main(String[] args) {
// TODO Auto-generated method stub
int n = 5;
PI(n);
}

public static void PI(int n) {
// base case;
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, 5, 4, 7, 6 };
int item = 4;
System.out.println(Index(arr, item, 0));
}

public static int Index(int[] arr, int item, int i) {
// base case;
if (i == arr.length) {
return -1;
}
if (arr[i] == item) {
return i;
}
return Index(arr, item, i+1);
}

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

public static int fac(int n) {
// base case;
if (n == 0) {
return 1;
}
int fn = fac(n-1);
return fn * n;
}

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) {
// base case;
if (n == 0) {
return ans;
}
return fac(n-1, ans * n);
}

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) {
// base case;
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) {
// base case;
if (n == 0 || n == 1) {
return n;
}
int f1 = fib(n-1);
int f2 = fib(n-2);
return f1 + f2;
}