RECURSION (ÖZYİNELEME)

Doç. Dr. İlhan AYDIN

Recursion (Özyineleme) nedir?

- Problemleri daha basit alt problemlere bölerek çözme yöntemi.
- Alt problemler de kendi içlerinde başka alt problemlere bölünebilir.
- Alt problemler çözülebilecek kadar küçülünce bölme işlemi durur.
- Özyinelemeli bir algoritma bir problemi çözmek için problemi iki veya daha fazla alt probleme bölen bir yöntemdir.

Animasyon

Faktöryel Hesabı

faktoryel(0) = 1;
faktoryel(n) = n*faktoryel(n-1);

faktoryel(3)

Animasyon

Faktöryel Hesabı

faktoryel(0) = 1;

faktoryel(n) = n*faktoryel(n-1);

faktoryel(3) = 3 * faktoryel(2)

Animasyon

Faktöryel Hesabı

```
faktoryel(0) = 1;
faktoryel(n) = n*faktoryel(n-1);
```

faktoryel(3) = 3 * faktoryel(2)

= 3 * (2 * faktoryel(1))

Animasyon

Faktöryel Hesabı

faktoryel(0) = 1;

faktoryel(n) = n*faktoryel(n-1);

faktoryel(3) = 3 * faktoryel(2)

= 3 * (2 * faktoryel(1))

= 3 * (2 * (1 * faktoryel(0)))

Animasyon

Faktöryel Hesabı

```
faktoryel(0) = 1;
faktoryel(n) = n*faktoryel(n-1);
```

faktoryel(3) = 3 * faktoryel(2)

= 3 * (2 * faktoryel(1))

= 3 * (2 * (1 * faktoryel(0)))

= 3 * (2 * (1 * 1)))

Animasyon

Faktöryel Hesabı

```
faktoryel(0) = 1;
faktoryel(n) = n*faktoryel(n-1);
```

faktoryel(3) = 3 * faktoryel(2)

= 3 * (2 * faktoryel(1))

= 3 * (2 * (1 * faktoryel(0)))

= 3 * (2 * (1 * 1)))

= 3 * (2 * 1)

Animasyon

Faktöryel Hesabı

faktoryel(0) = 1;

faktoryel(n) = n*faktoryel(n-1);

faktoryel(3) = 3 * faktoryel(2)

= 3 * (2 * faktoryel(1))

= 3 * (2 * (1 * faktoryel(0)))

= 3 * (2 * (1 * 1)))

= 3 * (2 * 1)

= 3 * 2

Animasyon

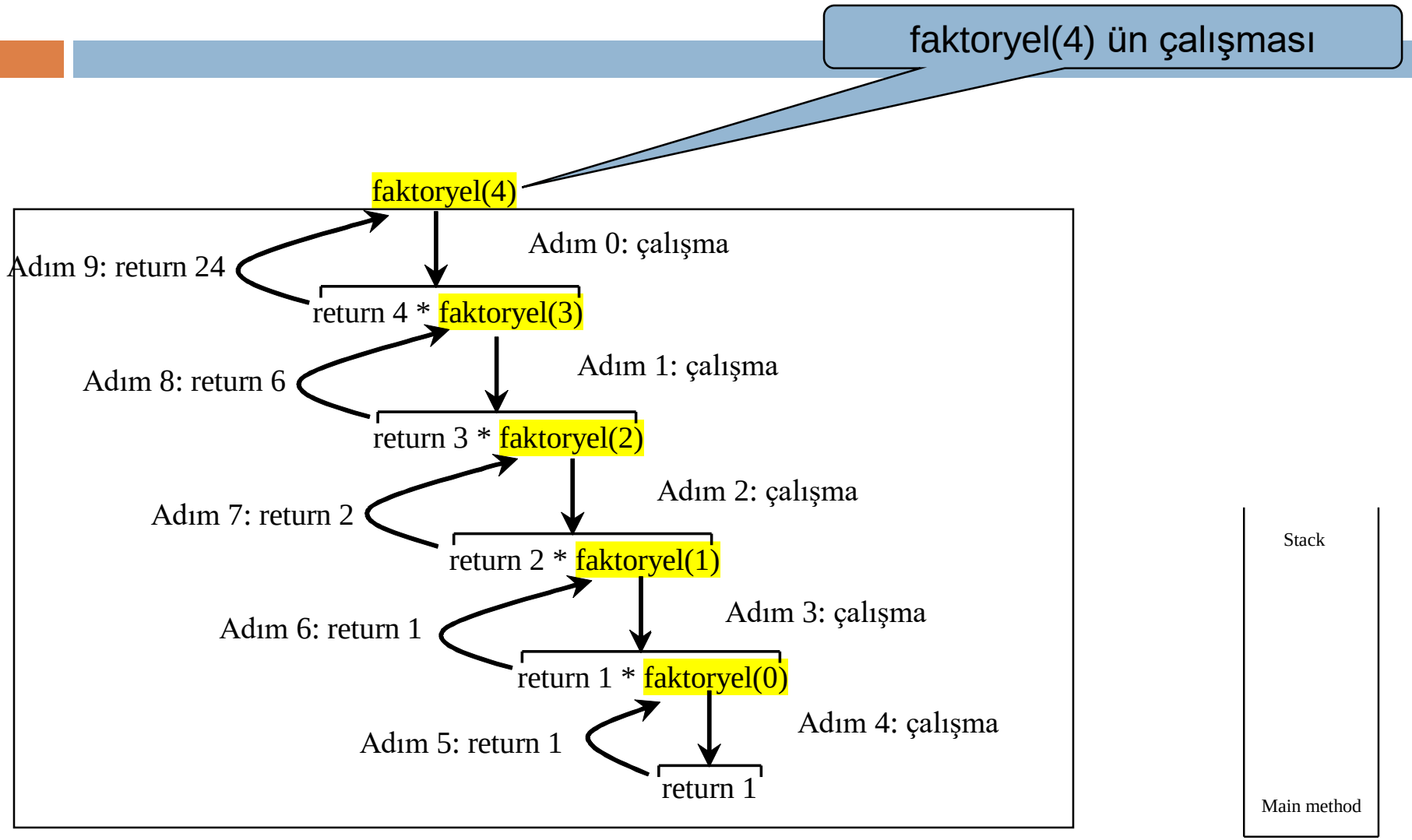
Faktöryel Hesabı

faktoryel(0) = 1;
faktoryel(n) = n*faktoryel(n-1);

faktoryel(3) = 3 * faktoryel(2)
= 3 * (2 * faktoryel(1))
= 3 * (2 * (1 * faktoryel(0)))
= 3 * (2 * (1 * 1)))
= 3 * (2 * 1)
= 3 * 2
= 6

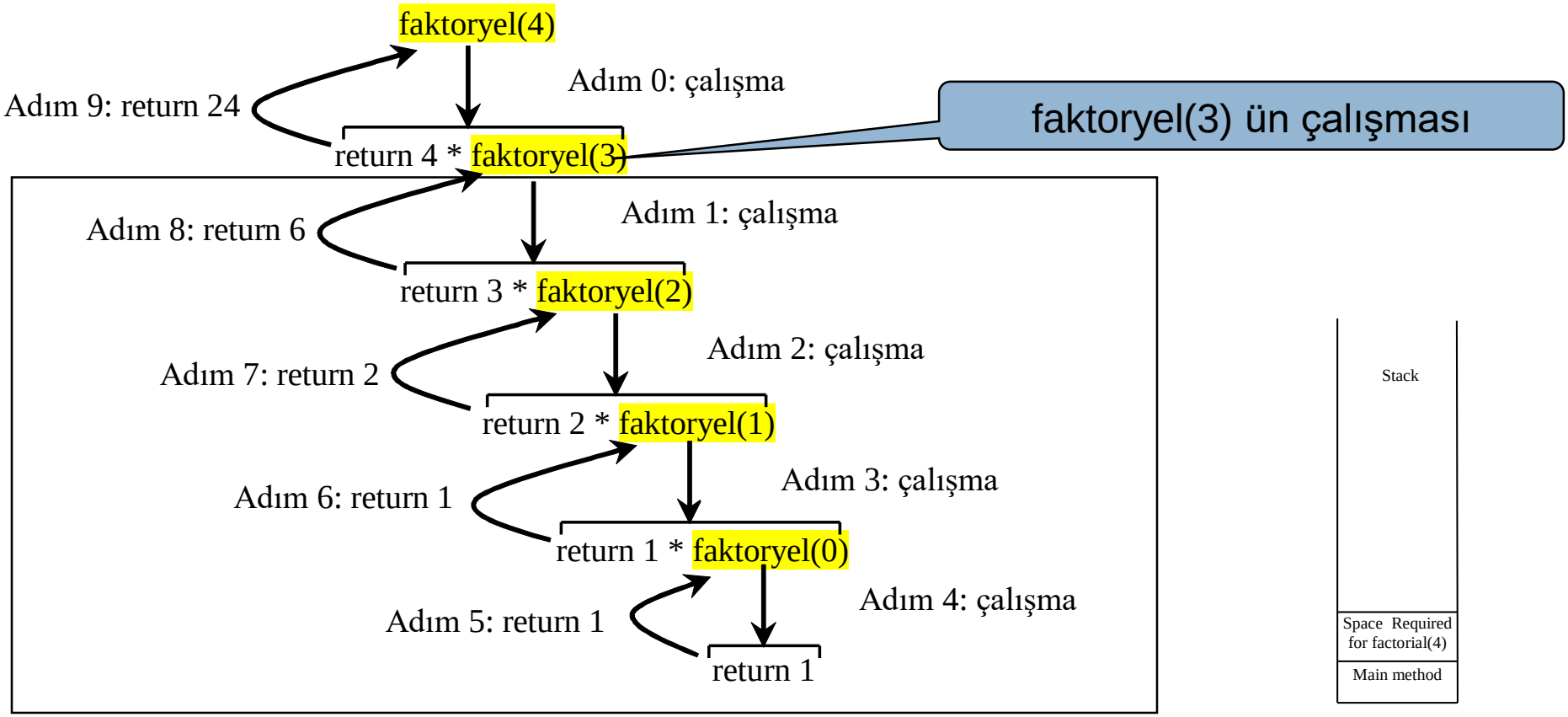
Animasyon

Recursive faktoryel



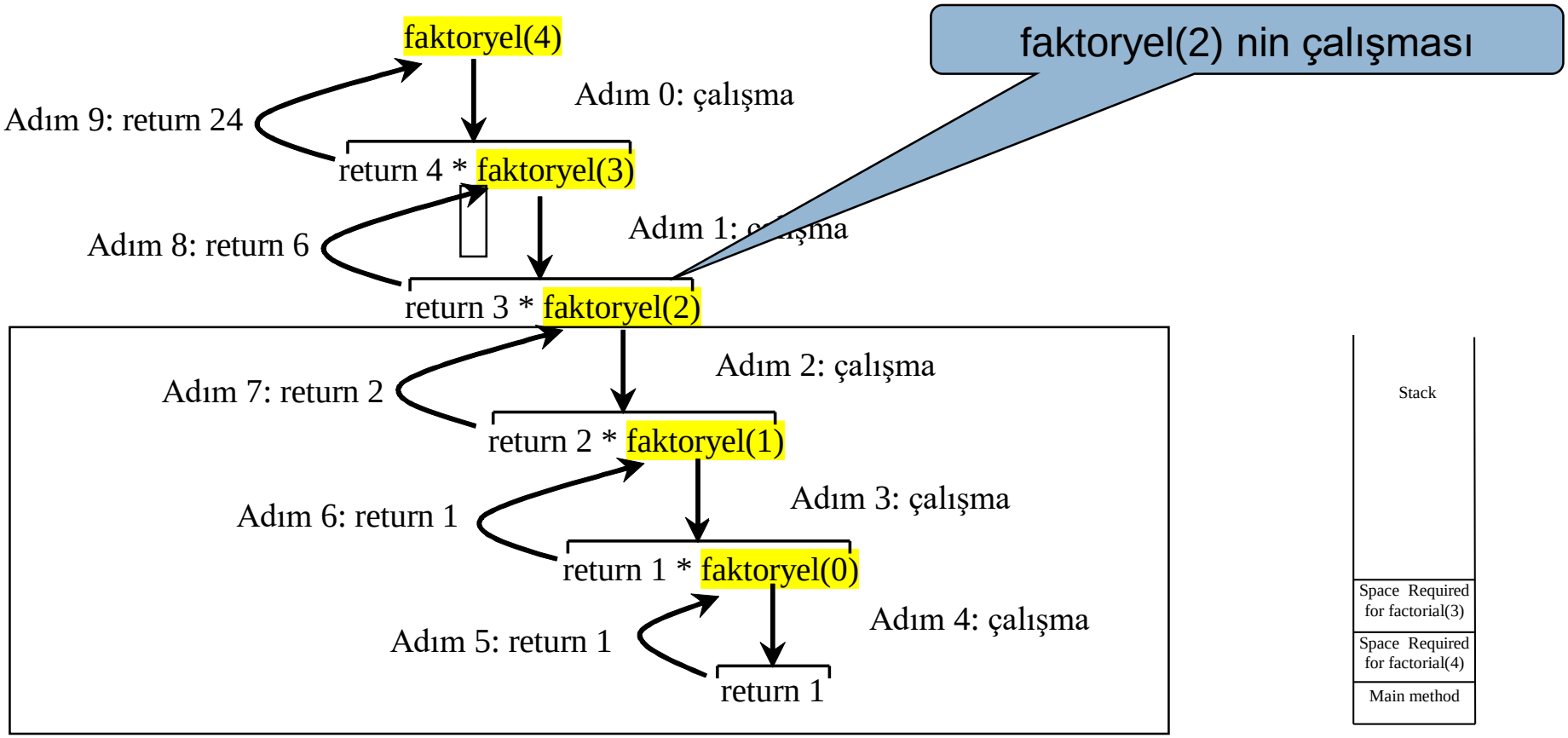
Animasyon

Recursif faktoryel



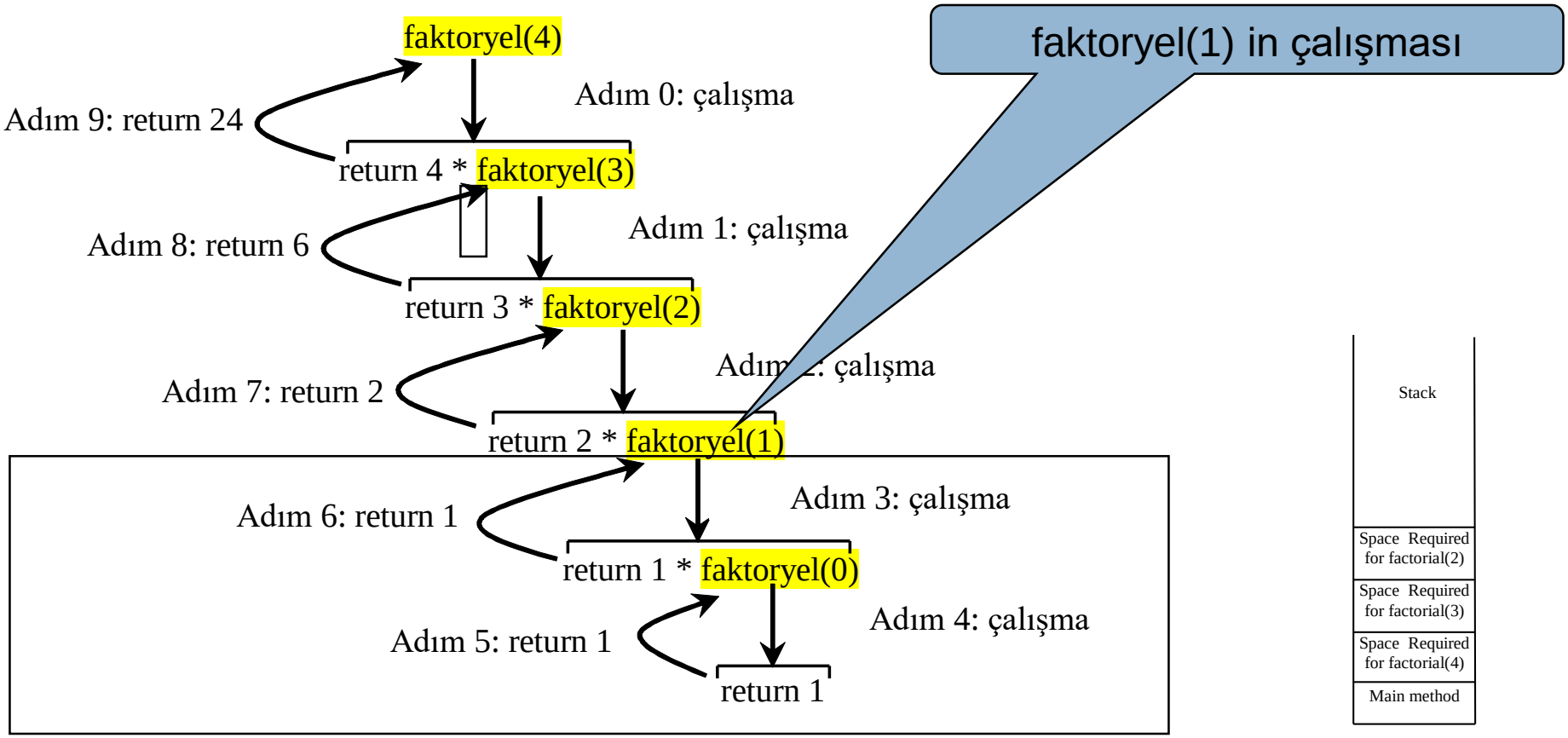
Animasyon

Recursive faktoryel



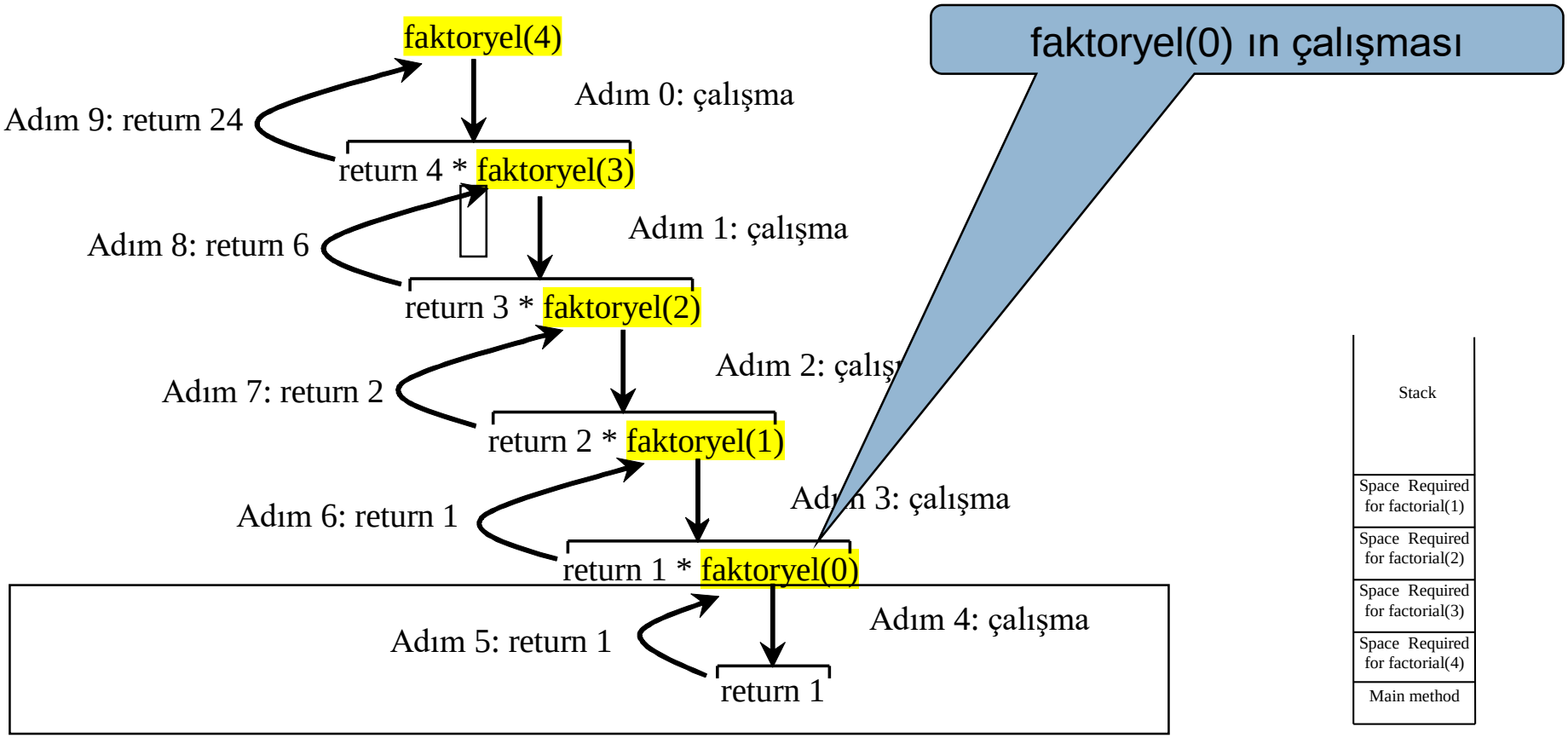
Animasyon

Recursive faktoryel



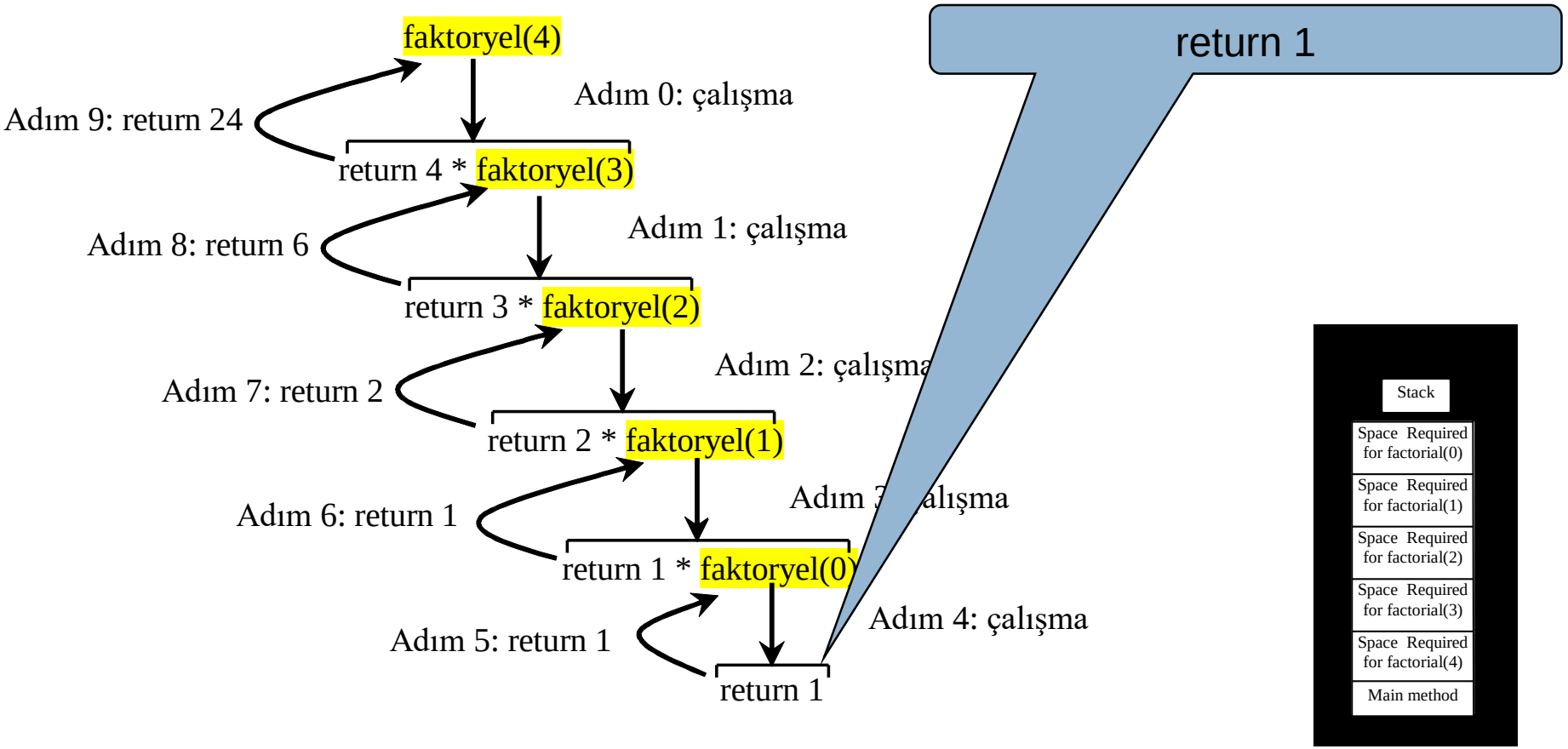
Animasyon

Recursive faktoryel



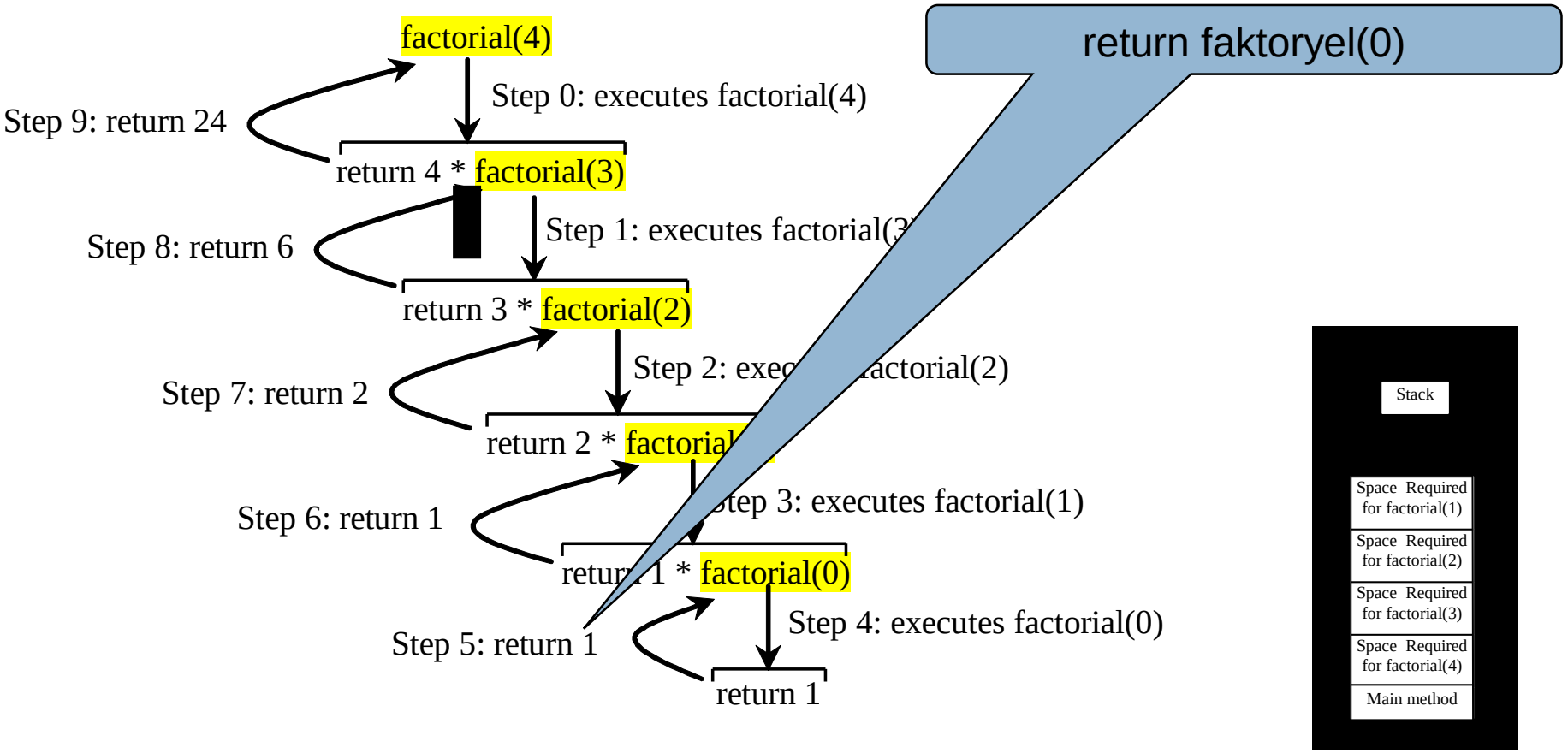
Animasyon

Recursive faktoryel



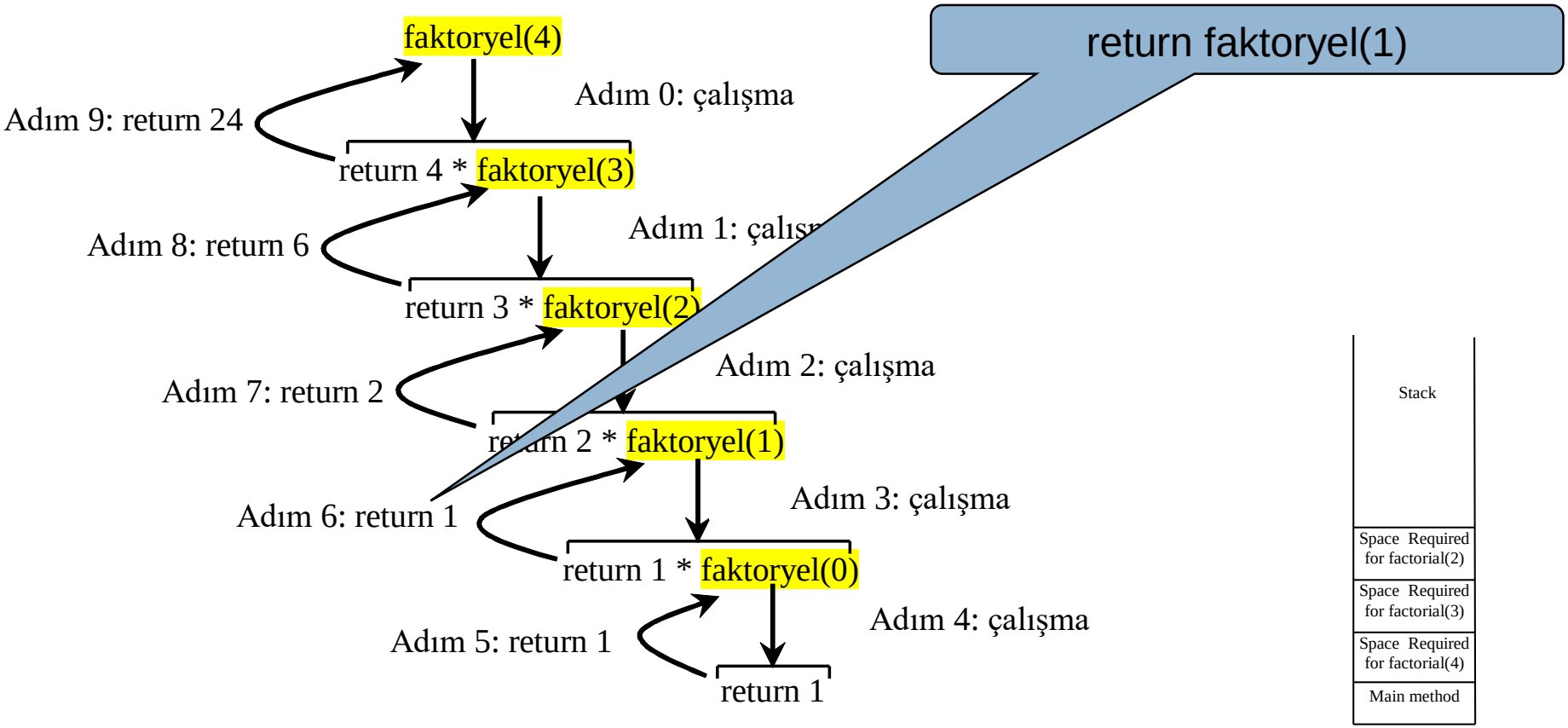
Animasyon

Recursive faktoryel



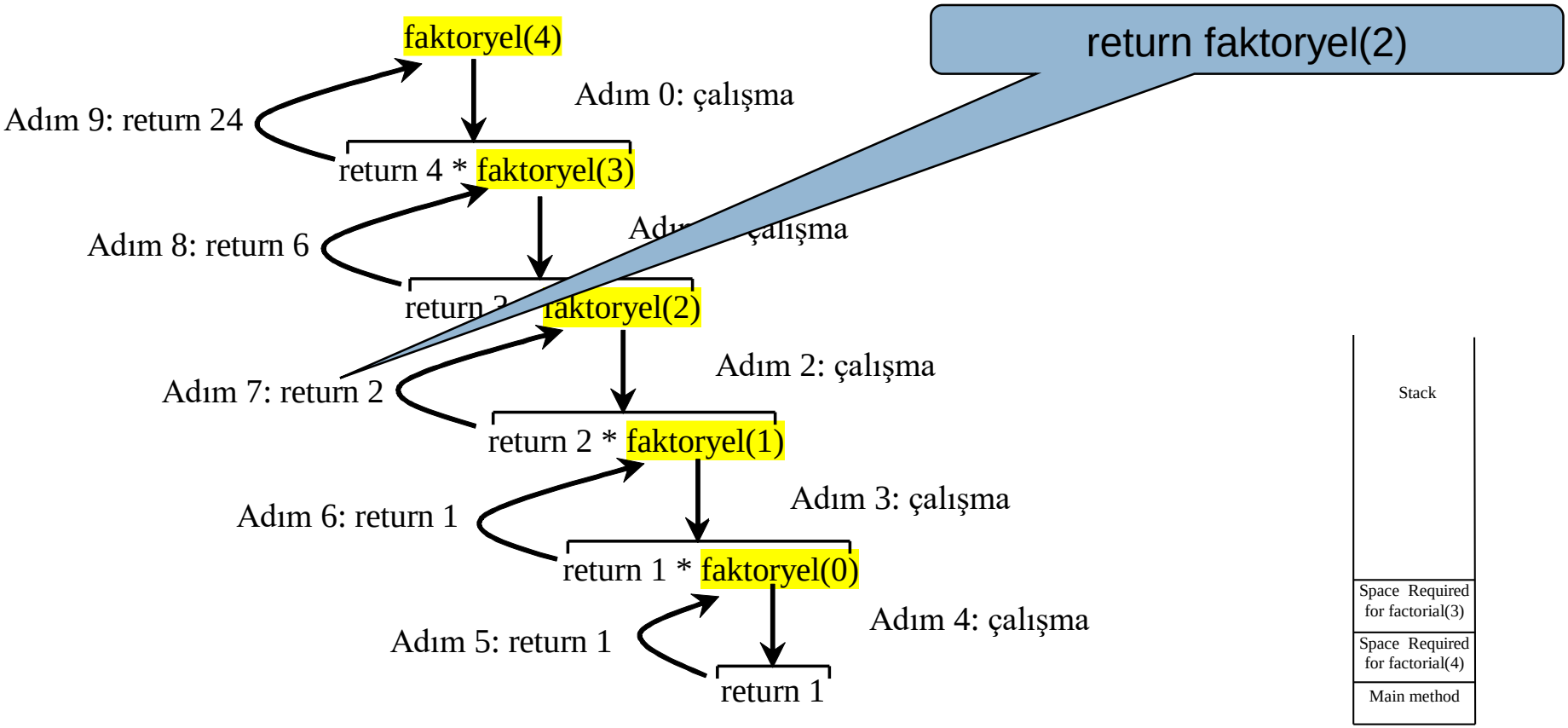
Animasyon

Recursive faktoryel



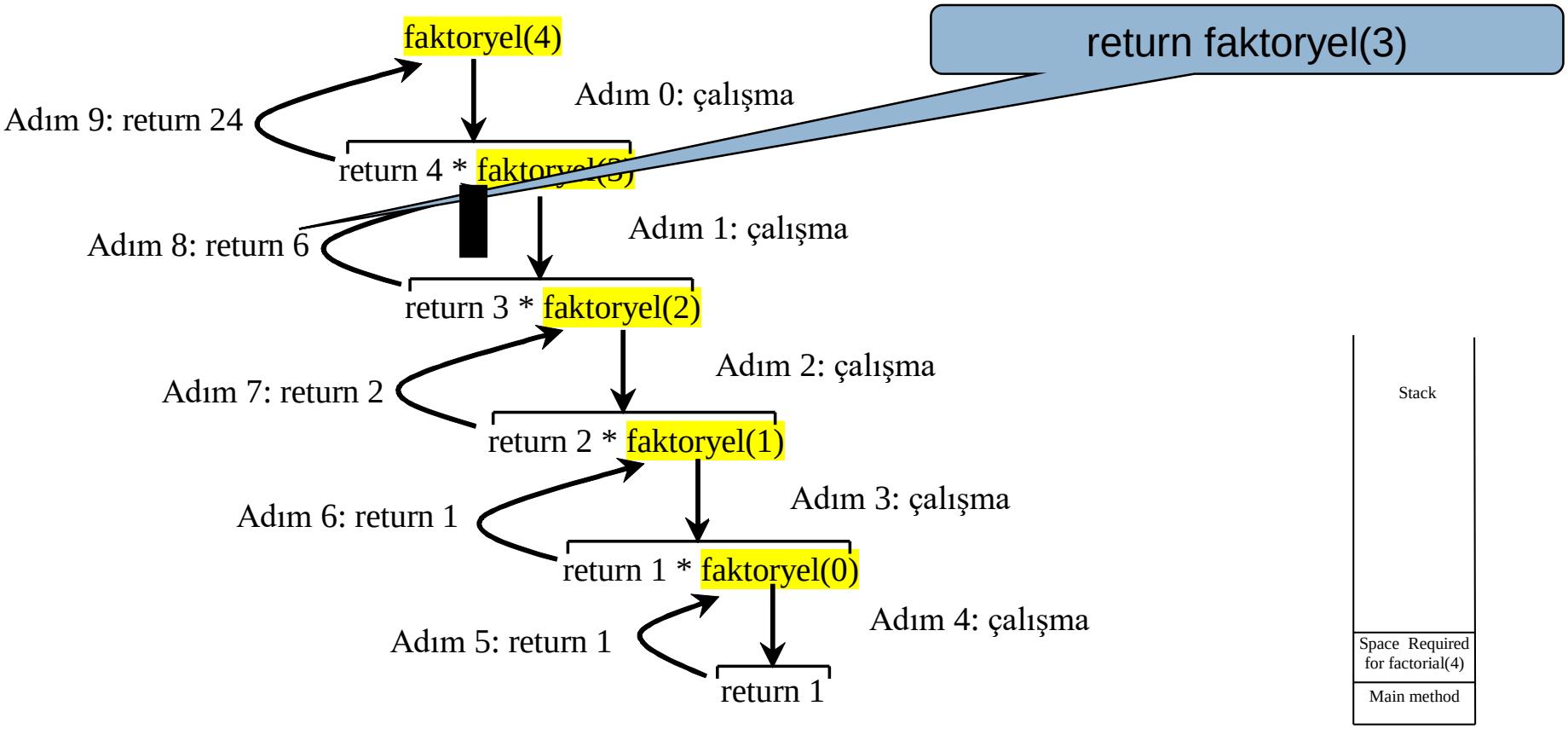
Animasyon

Recursive faktoryel



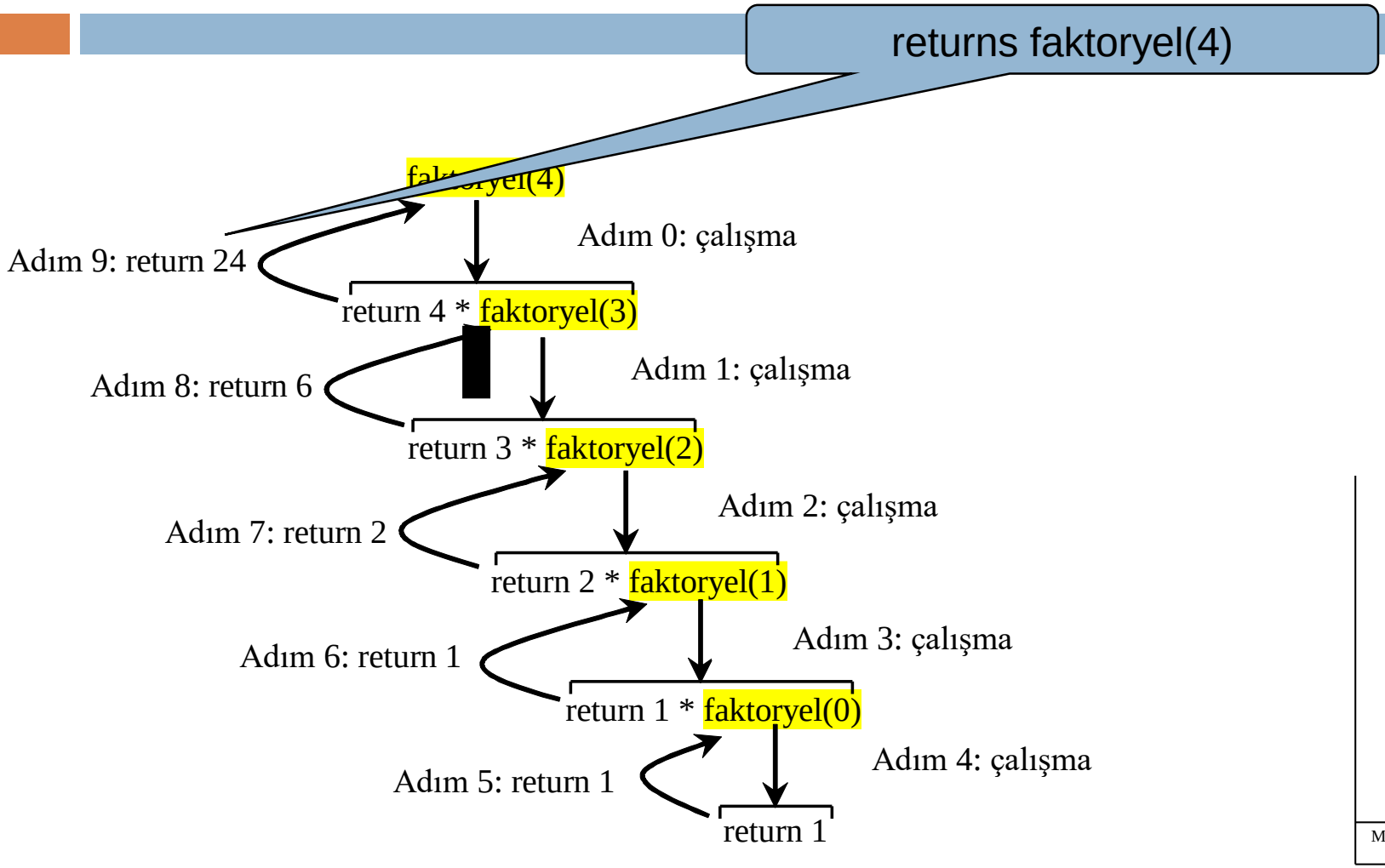
Animasyon

Recursive faktoryel

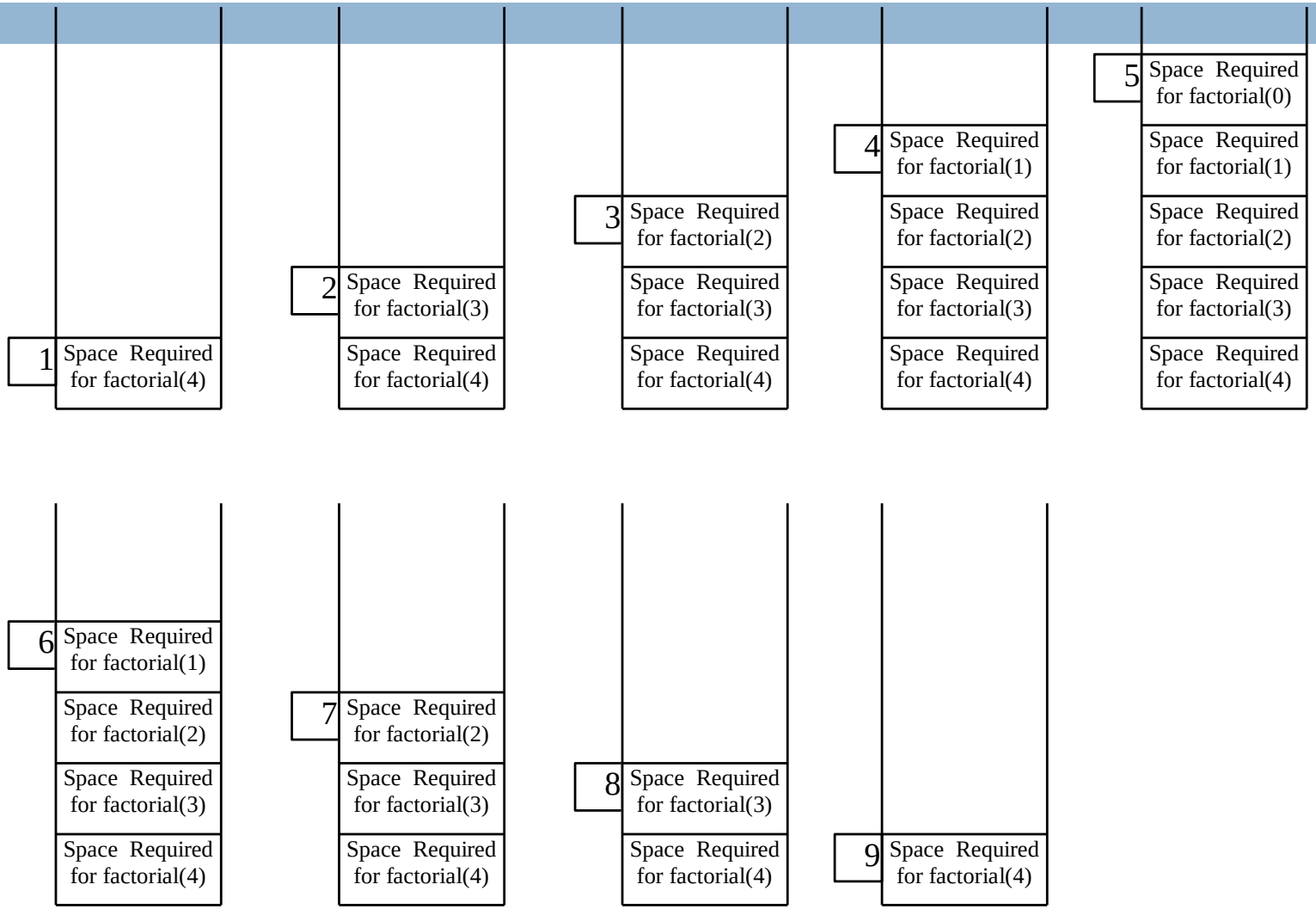


Animasyon

Recursive faktoryel



faktoryel(4) Stack



Basamak Sayısı

- Ozyinelemeli Tanim

  basamak(n) = 1 -> if (–9 <= n <= 9)

  1 + basamak(n/10) -> degilse

- Ornek

  basamak(321) =

  1 + basamak(321/10) = 1 +basamak(32) =

  1 + [1 + basamak(32/10)] =

  1 + [1 + basamak(3)] =

  1 + [1 + (1)] =

  3

Basamak Sayısı

```
int basamakSayisi(int n) {
   if ((-10 < n) && (n < 10))
      return 1
   else
      return 1 +
       basamakSayisi (n/10);
}
```

Özyineleme

- f(x) problemini çözmek istiyorsak fakat direkt olarak çözemiyorsak
- Varsayalım y`nin x`den küçük herhangi bir değeri için f(y)'yi çözebiliyoruz
- f(x)`i çözmek için f(y)`yi kullanırız
- Bu yöntemin çalışabilmesi için f(x)`in direkt olarak hesaplanabildiği en az bir x değerinin olması gerekir. (e.g. *taban durumu*)

Bir Sayının kuvvetini hesaplama

```
static int power(int k, int n) {
  // k`nın n. üssü
  if (n == 0)
    return 1;
  else
    return k * power(k, n - 1);
}
```

Toplam Hesaplama

```
public int sum(int num) {
        int result;
        if (num == 1) {
            result = 1;
        } else {
            result = num + sum(num - 1);
        }
        return result;
    }
```

Faktoriyel

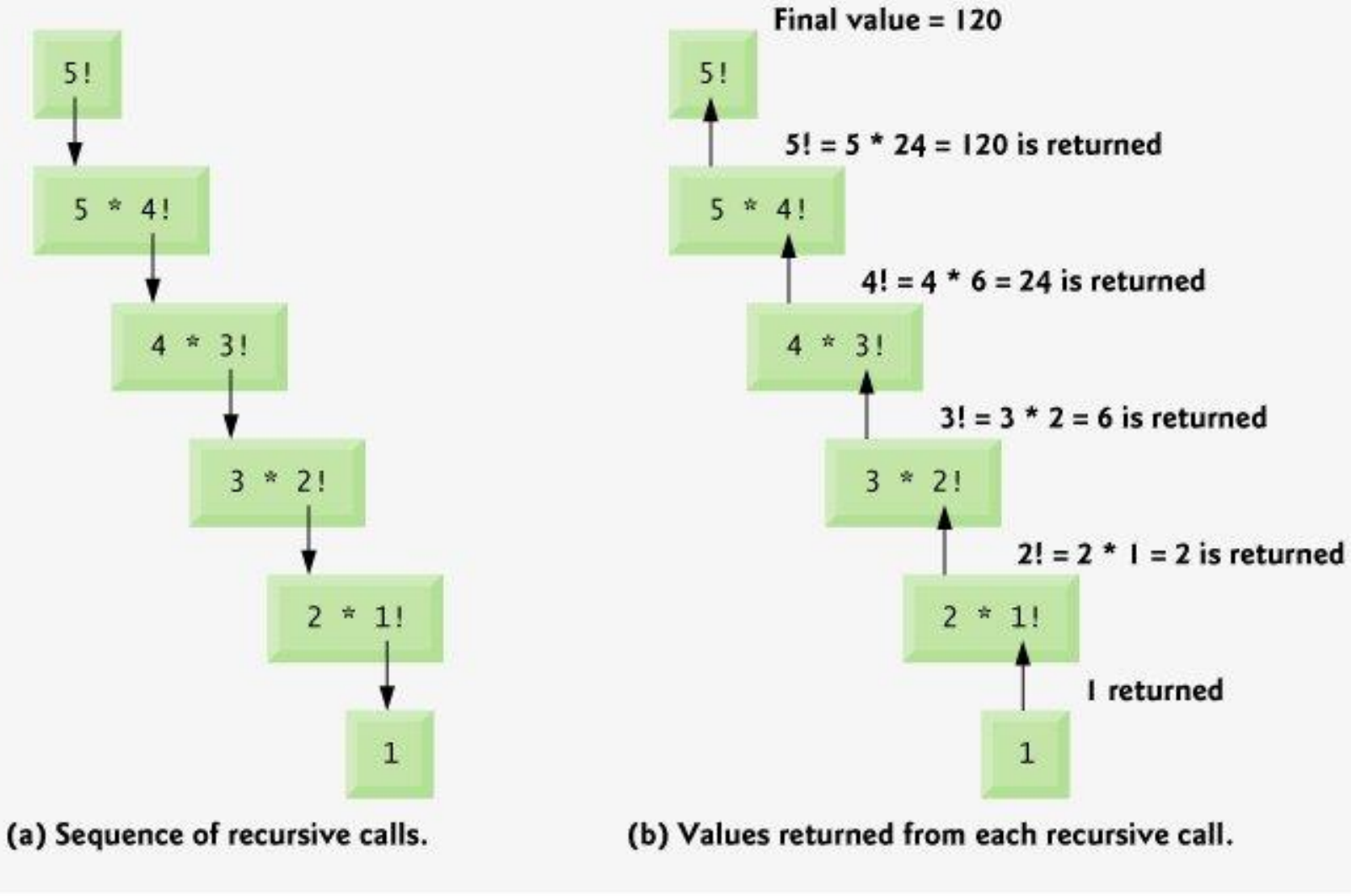
- 5! = 5*4*3*2*1
  4! = 4*3*2*1
  3! = 3*2*1
  2! = 2*1
  1! = 1

```
int faktoriyel = 1;
for ( int sayac = 5; sayac>= 1; sayac-- )
faktoriyel = faktoriyel *sayac;
```

Özyinelemeli Faktöriyel

```
public int faktoriyel(int N) {
    if (N == 0) return 1;
    return N*faktoriyel(N-1);
}
```

Recursive faktoryel



```
public double faktoriyelGoster(int n, int sayac) {
    double fakt;
    if (n <= 0) {
        System.out.println("Taban Duruma ulasti");
        fakt = 1;
    } else {
        sayac++;
        System.out.println(sayac + ". program
cagiriyor ");
        fakt = n * faktoriyelGoster(n - 1, sayac);
    //System.out.println("Faktoriyel = " + fakt);
    }
    System.out.println(sayac + ". programdan
Cikiyor ");
    sayac--;
    return fakt;
}
```

Faktoriyel programında ozyinelemeli metodların çağrılması

1. program cagiriyor
2. program cagiriyor
3. program cagiriyor
4. program cagiriyor
5. program cagiriyor

Taban Duruma ulasti

5. programdan Cikiyor
5. programdan Cikiyor
4. programdan Cikiyor
3. programdan Cikiyor
2. programdan Cikiyor
1. programdan Cikiyor

Fibonacci Sayıları

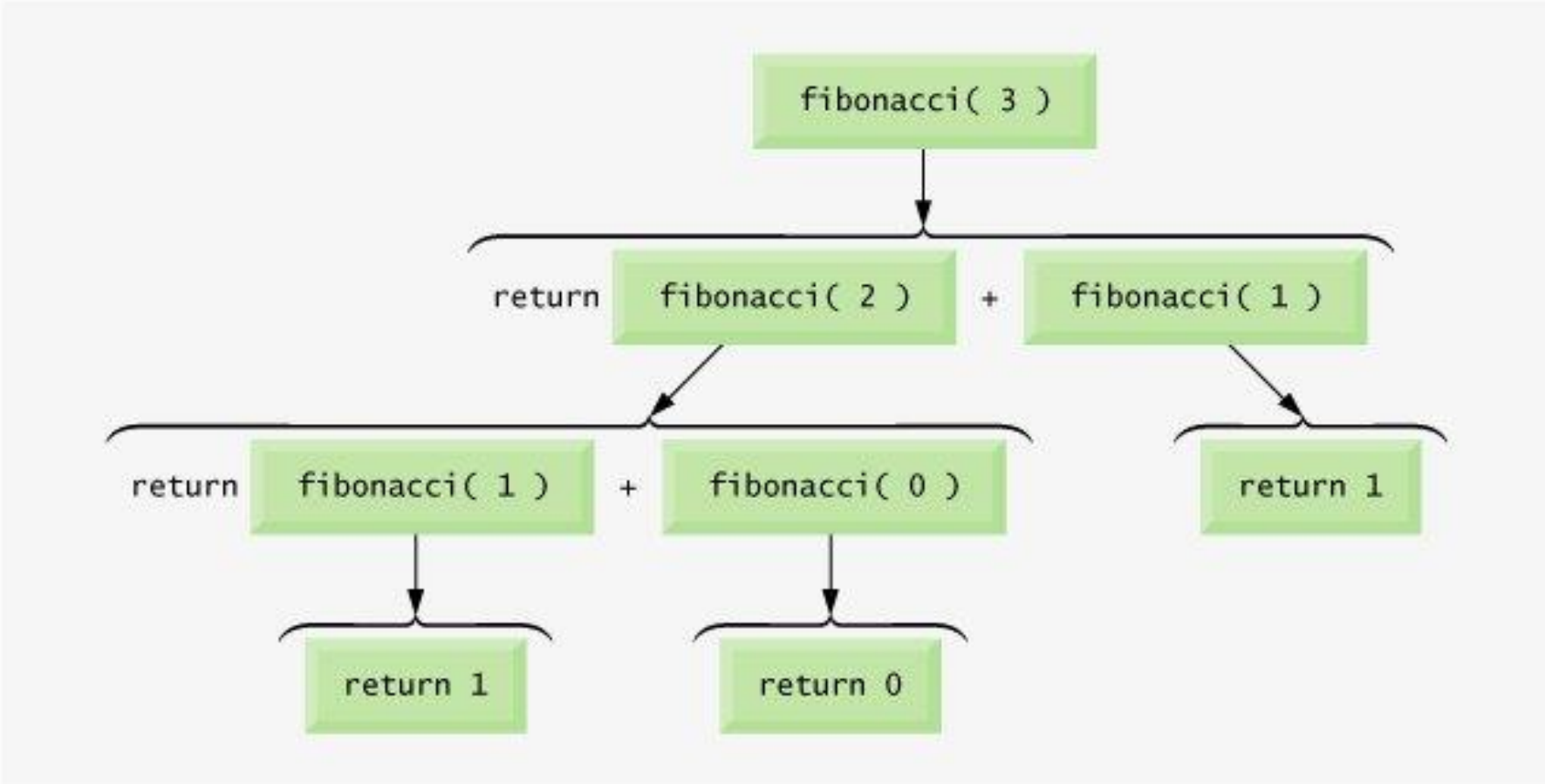
$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

$$\begin{aligned} \text{fibonacci}(0) &= 0 \\ \text{fibonacci}(1) &= 1 \\ \text{fibonacci}(n) &= \text{fibonacci}(n-1) + \text{fibonacci}(n-2) \end{aligned}$$

Fibonacci Java

```
public int fibonacci(int sayi) {
    if ( ( sayi == 0 ) || ( sayi == 1 )
        return sayi;
    else
        return fibonacci( sayi - 1 ) + fibonacci( sayi - 2 );
}
```

Fibonacci Metodunun Çalışması



Örnek: Kendisine parametre olarak gelen dizgiyi tersten yazan rekursif metot

```
1  import java.util.Scanner;
2  public class uygulama2 {
3      public static void tersyaz(String S){
4          if(S.length()>0){
5          System.out.print(S.substring(S.length()-1,S.length()));
6          tersyaz(S.substring(0, S.length()-1));
7          }
8      }
9      public static void main(String[] args) {
10         Scanner klavye=new Scanner(System.in);
11          System.out.println("Dizgi gir");
12         String dizgi=klavye.nextLine();
13         tersyaz(dizgi);
14     }
15 }
```

Örnek: Bir dizgede bir karakterin kaç kez geçtiğini bulan rekürsif metot

```
1  import java.util.Scanner;
2  public class uygulama2 {
3      public static int count(String str,char a){
4          if(str.length()<1)
5              return 0;
6          else if(str.substring(str.length()-1, str.length()).charAt(0)==a)
7              return 1+count(str.substring(0, str.length()-1),a);
8          else
9              return 0+count(str.substring(0, str.length()-1),a);
10
11     }
12     public static void main(String[] args) {
13         Scanner klavye=new Scanner(System.in);
14          System.out.println("Dizgi ve aranan karakteri gir:");
15         String dizgi=klavye.nextLine();
16         char b=klavye.next().charAt(0);
17          System.out.println(count(dizgi,b));
18     }
19 }
```