

Recursión

Estructura de Datos y Algoritmos - TICS311

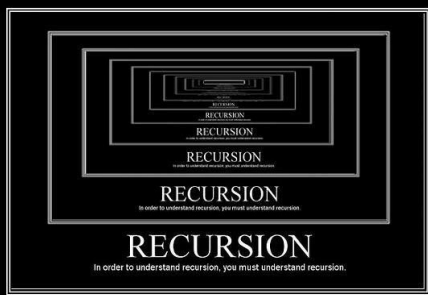
1

¿Qué es recursión?

En ciencias de la computación, un algoritmo recursivo es aquel que se llama a sí mismo.

Estructura de Datos y Algoritmos - TICS311

2



Estructura de Datos y Algoritmos - TICS311

RECURSION

In order to understand recursion, you must understand recursion.

3

Función recursiva

```
function hola():  
    print("hola")  
    hola()
```

4

Función recursiva

```
function hola():  
    print("hola")  
    hola()
```

Screen

hola

5

Función recursiva

```
function hola():  
    print("hola")  
    hola()
```

Screen

hola

6

Función recursiva

```
function hola():  
  print("hola")  
  function hola():  
    print("hola")  
    hola()
```

Screen

```
hola  
hola
```

7

Función recursiva

```
function hola():  
  print("hola")  
  function hola():  
    print("hola")  
    hola()
```

Screen

```
hola  
hola
```

8

Función recursiva

```
function hola():  
  print("hola")  
  function hola():  
    print("hola")  
    hola()
```

Screen

```
hola  
hola  
hola
```

9

Función recursiva

```
function hola():
    print("hola")
function hola():
    print("hola")
function hola():
    print("hola")
    hola()
    hola()
```

Screen

hola

hola

hola

•

•

•

¿Cual es el problema con esta función?

10

Función recursiva

- ◊ Una Función recursiva tiene dos partes:
 - ◊ Una o más condiciones de término
 - ◊ Una o más llamadas recursivas

Estructura de Datos y Algoritmos - TICS2011

11

11

Ejemplo

```
function suma(a,b):
    if (a==0):
        return b
    if (b==0):
        return a
    return suma(a+1,b-1)
```

12

Ejercicios (Use recursión!)

- ♦ Haga una función en C que permita calcular la suma de todos los números naturales desde 1 hasta n, donde n es un parámetro.
- ♦ Haga una función que imprima por pantalla los dígitos de un número n de manera invertida. Por ejemplo si el parámetro es 5476, debe imprimir por pantalla 6754.
- ♦ Escriba un programa que muestre por pantalla el n-ésimo número (partiendo desde el cero) de la secuencia de Fibonacci:

0-1-1-2-3-5-8-13-21-34...

13

Complejidad temporal

Recursiones

Estructura de Datos y Algoritmos - TICS311

14


14

```
function fact(N)
    if(N==0)
        return 1
    return N*fact(N-1)
```

Estructura de Datos y Algoritmos - TICS311

15

15



```


function fact(N)
  if(N==0) → C1
    return 1
  return N*fact(N-1) → C2+T(N-1)

```

Estructura de Datos y Algoritmos - TICS011

16

16



```

function fact(N)
  if(N==0) → C1
    return 1
  return N*fact(N-1) → C2+T(N-1)


```

La complejidad temporal de un algoritmo recursivo también puede ser descrita usando una ecuación recursiva.

Estructura de Datos y Algoritmos - TICS011

17

17



$T(N) = T(N-1) + C_3$


```

function fact(N)
  if(N==0) → C1
    return 1
  return N*fact(N-1) → C2+T(N-1)

```

$T(0) = C$

18




$T(N) = T(N-1) + C_3$
 $T(N) = (T(N-1) + C_3) + C_3$
 $T(N) = ((T(N-1) + C_3) + C_3) + C_3$
 $T(0) = C$

```

function fact(N)
  if(N==0) → C1
  return 1
  return N*fact(N-1) → C2+T(N-1)
  
```

19




$T(N) = T(N-1) + C_3$
 $T(N) = (T(N-1) + C_3) + C_3$
 $T(N) = ((T(N-1) + C_3) + C_3) + C_3$
 $T(0) = C$

```

function fact(N)
  if(N==0) → C1
  return 1
  return N*fact(N-1) → C2+T(N-1)
  
```

20



$T(N) = T(N-1) + C_3$
 $T(N) = (T(N-1) + C_3) + C_3$
 $T(N) = ((T(N-1) + C_3) + C_3) + C_3$
 \vdots
 $T(N) = T(N-k) + kC_3$
 $T(N) = T(N-N) + NC_3$
 $T(0) = C$

```

function fact(N)
  if(N==0) → C1
  return 1
  return N*fact(N-1) → C2+T(N-1)
  
```

21

T(N)=T(N-1) + C₃

T(N)= (T(N-1) + C₃) + C₃

T(N)= ((T(N-1) + C₃) + C₃) + C₃

⋮

T(N)= T(N-k) + kC₃

T(N)= T(N-N) + NC₃

T(0)= C

function fact(N)
if(N==0) → C₁
return 1
return N*fact(N-1) → C₂+T(N-1)

T(N)= C + NC₃

22

T(N)= C + NC₃

Para la función factorial recursiva

- T(N) crece linealmente
- T(N) = O(N)
- T(N) = Ω(N)
- T(N) = Θ(N)

Estructura de Datos y Algoritmos - TICS311

23

23

Ejercicio

$$a^b = \begin{cases} a * a^{(b-1)} & , \text{ if } b > 0 \\ 1 & , \text{ if } b = 0 \end{cases}$$

function exp(a,b)
if(b==0)
return 1
return a*exp(a,b-1)

Estructura de Datos y Algoritmos - TICS311

24

24
