UNIVERSIDAD NACIONAL DE HURLINGHAM

Estructuras de Datos

Profesor

Sergio Gonzalez

Unidad 3: Recursividad

Profesor

Sergio Gonzalez

Recursividad

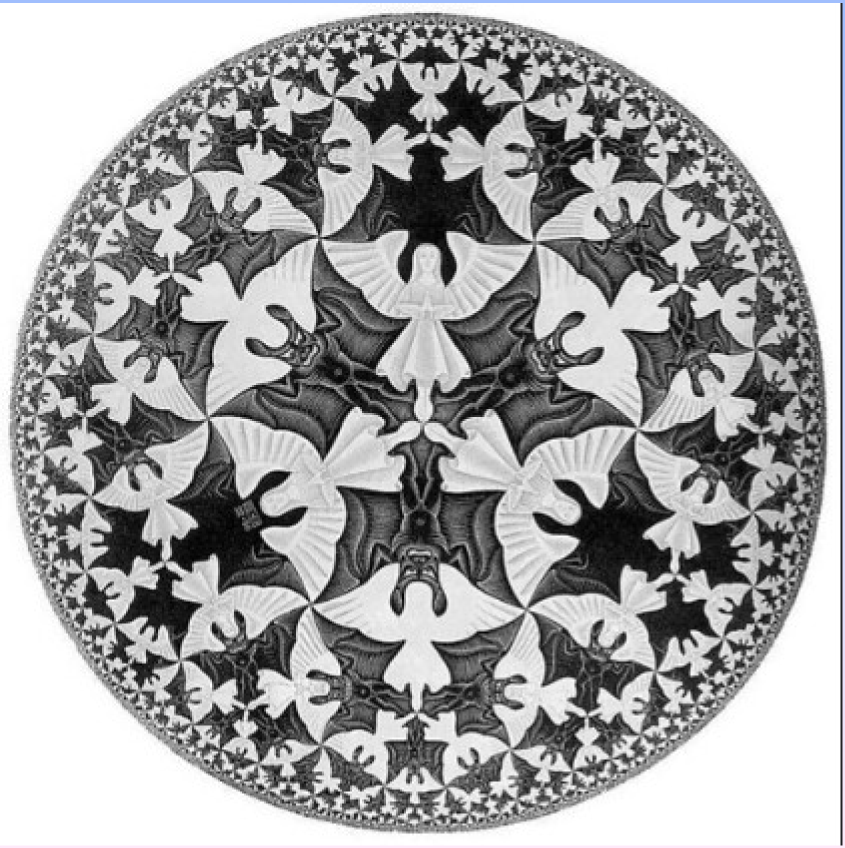
- Alternativa a la repetición (soluciones iterativas)
- Siempre existen ambas soluciones, complejidad dependiendo del problema
- Problemas con ‘estructura de recurrencia’

Estructura de recurrencia

- Tenemos un problema A
- Se divide en dos partes B y C mas pequeñas
- Si una de esas partes (por ejemplo B) es idéntica a A
- Se conoce solución explícita para caso simple
- Entonces el problema es recursivo, porque la resolución de B se puede dividir igual que la de A

Ejemplo

- Calculo del factorial
  - N! = N (N - 1)!
  - (N - 1)! = (N – 1) (N - 2)!



Programación de solución recursiva

Tenemos una función que se llama a si misma

Programación de solución recursiva

2 Partes principales:

- **Caso general:** La autollamada a una o mas versiones mas chicas del problema y algunas cosas mas. Las llamadas recursivas deben converger hacia el caso base.
- **Caso base:** Solución de caso simple conocida. Para que sea finito, debe haber una condición de corte (sin autollamada)

Programación de solución recursiva

- Obtener definición exacta del problema
- Determinar el tamaño del problema general
- Obtener el/los casos base (no recursivos)
- Resolver el caso general en términos de uno mas chico (llamada recursiva)

Ejemplos

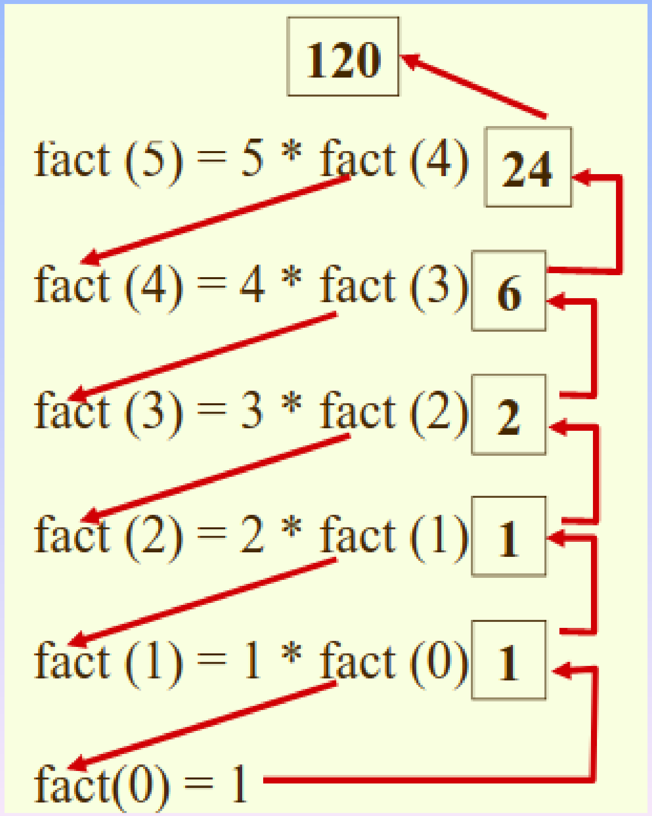
- Ejemplo 1: Calculo del factorial
- Solución en pseudocódigo

```
función Factorial (n)
inicio
    si n = 0
        entonces Factorial ←1
        sino Factorial ← n * Factorial (n-1)
    fin_si
fin
```

Ejemplos

- Ejemplo 1: Calculo del factorial de 3, descripción de llamadas apiladas:
  - factorial(3) -> 3 * factorial(2)
  - factorial(2) -> 2 * factorial(1)
  - factorial(1) -> 1 * factorial(0)
  - factorial(0) <- 1

- Ejemplo 1: Calculo del factorial de 5, descripción de llamadas apiladas:



Ejemplos 2: Suceción de Fibonacci

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕

F(1), F(2), F(3), F(4), F(5), F(6), F(7), F(8), F(9), F(10), F(11), ...

UNIVERSIDAD NACIONAL DE HURLINGHAM

Ejemplos 2: Suceción de Fibonacci

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

F(1), F(2), F(3), F(4), F(5), F(6), F(7), F(8), F(9), F(10), F(11), ...

Fibonacci(N) = Fibonacci(N - 1) + Fibonacci(N – 2)

- Fibonacci(N - 2) = Fibonacci(N - 3) + Fibonacci(N - 4)

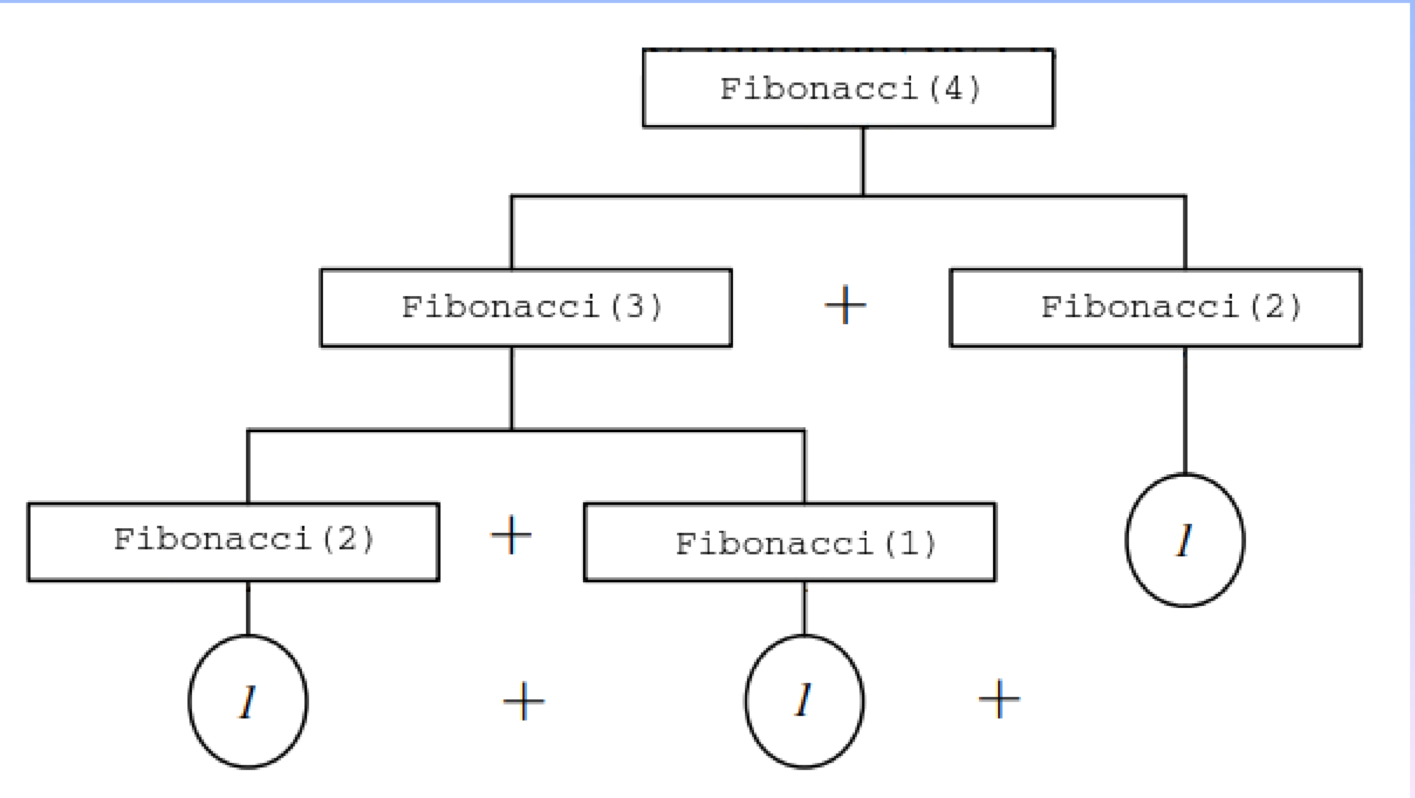
- Fibonacci(N - 1) = Fibonacci(N - 2) + Fibonacci(N - 3)

Ejemplos 2: Suceción de Fibonacci

- Solución en pseudocódigo

```
función FIBONACCI(n)
inicio
    si (n=1) o (n=2)
        entonces
            FIBONACCI ← 1
        sino
            FIBONACCI ← FIBONACCI (n-2) + FIBONACCI (n-1)
    fin_si
fin_función
```

- Calculo del 4 número de la suceción de Fibonacci, descripción de llamadas apiladas:



Ejemplos 2: Suceción de Fibonacci

Problema con Sucesión de Fibonacci:

– Fibonacci(N) = Fibonacci(N - 1) + Fibonacci(N – 2)

- Fibonacci(N - 1) = **Fibonacci(N - 2)** + **Fibonacci(N - 3)**
- **Fibonacci(N - 2)** = **Fibonacci(N - 3)** + Fibonacci(N - 4)

Ejemplos 2: Suceción de Fibonacci

Problema con Sucesión de Fibonacci:

- Cantidad exponencial de llamadas recursivas