



Resolución de Problemas y Algoritmos

Clase 18
Soluciones a problemas usando recursión:
planteos recursivos,
funciones y procedimientos recursivos



Dr. Alejandro J. García
 http://cs.uns.edu.ar/~ajg



Departamento de Ciencias e Ingeniería de la Computación
 Universidad Nacional del Sur
 Bahía Blanca - Argentina

Solución para contestar preguntas

- El martes 4 de junio no habrá clase teórica.
- Estamos a 22 días del examen, falta el examen en máquina y muchos aún pueden promocionar.

¿Alguna pregunta?

Contestar preguntas

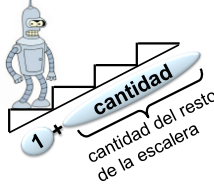
- (Caso Base) Si hay un solo alumno con preguntas: escuchar y contestar sus preguntas.
- (Caso General) Si hay más de un alumno que quiere preguntar, elegir uno, escuchar sus preguntas y contestarlas. Luego contestar las preguntas del resto de los alumnos.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

(Repaso) Algoritmos recursivos

- Un planteo recursiva **será válido**, si:
 - (a) **existe un caso base** que **no** se define en términos de si mismo, y en el caso general,
 - (b) la **referencia a si mismo es relativamente más sencilla o reducida** que el caso considerado.

Ejemplo:
Planteo: Cantidad de escalones
 (CB) si no hay escalones:
 la cantidad es 0 (cero)
 (CG) si hay escalones:
 sube un escalón y
 la cantidad es 1 + la cantidad del resto de la escalera



Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

Metodología propuesta

1. Identificar **ejemplos significativos** que ayuden a entender el problema y su solución.
2. Realizar un **planteo recursivo** en el cual se distinga el "caso base", y el "caso general" (donde se define en términos de si mismo pero para una instancia más simple/reducida/menor).
3. **Verificar** que el planteo es correcto (con alguno de los ejemplos significativos).
4. Determinar si se realizará una **función** o un **procedimiento recursivo**, e implementarlo en Pascal.
5. Realizar la **traza** de la primitiva en Pascal.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

Problema propuesto: cantidad de dígitos

Escriba un planteo recursivo y luego una función (o procedimiento) que respete ese planteo para contar la cantidad de dígitos de un número entero.
 Ejemplos: 12345 (5 díg) -14 (2 díg) 9 (1 díg) 0 (1 díg)

Planteo: Contar cantidad de dígitos de N
Caso base: si N tiene un único dígito ($N \div 10 = 0$) entonces la cantidad es 1
Caso general: si N tiene más de un dígito, entonces la cantidad es 1 + la cantidad de dígitos de N sin su último dígito ($N \div 10$)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

Tarea

- Durante la clase, en el pizarrón, fue hecha una función, pero para practicar puede hacer dos versiones: una función y un procedimiento.
- En una transparencia más abajo hay tres versiones de una función para factorial que muestran diferentes formas de implementar
- Implemente un programa y ejecute.
- Puede agregar algunos "write" en su primitiva recursiva para "ver" una traza automática de como se modifican sus variables.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

Problema propuesto: cantidad de elementos

Escriba un planteo recursivo y luego una función (o procedimiento) que respete ese planteo para contar la cantidad de elementos de un archivo.
Ejemplos: 1 2 3 4 (4 elem.) a b (2 elem) vacío (0 elem.)

Planteo: Contar cantidad de elementos de un archivo
Caso base: si el archivo está vacío entonces la cantidad es 0 (cero)
Caso general: si el archivo no está vacío, entonces la cantidad es 1 + la cantidad de elementos del archivo sin su primero elemento.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

Tarea

- Durante la clase, en el pizarrón, fue hecha una función, pero para practicar puede hacer dos versiones: una función y un procedimiento.
- En ambos casos tenga cuidado donde realiza “assign” y “reset” del archivo.
- Sugerencia: Pase en la máquina y vea la ejecución.
- **Pregunta teórica:** ¿necesita hacer una función diferente para cada tipo diferente de archivo? Escriba su respuesta y consulte sus dudas.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

Tarea propuesta

A continuación hay tres problemas propuestos con sus correspondientes planteos.
Para practicar se propone:

- 1) sin mirar el planteo propuesto intente primero escribir su propio planteo para resolver el problema y luego comparar con el propuesto.
- 2) escriba una función o procedimiento que siga su planteo
- 3) pase en la máquina
- 4) consulte sus dudas

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

Problema propuesto: cantidad de letras

Escriba un planteo recursivo y luego una función (o procedimiento) que respete ese planteo para contar la cantidad de letras ingresadas por teclado, de una secuencia finalizada en un punto.
Ejemplos: 1 2 3. (5 elem.) a. (1 elem) . (0 elem.)

Planteo: Contar cantidad de letras
Caso base: si leo un “.” entonces la cantidad es 0
Caso general: si leo algo que no es un “.”, entonces la cantidad es 1 + la cantidad de letras de la secuencia sin su primero elemento.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

Problema propuesto: dígito presente

Escriba un planteo recursivo y luego una función (que respete ese planteo) que retorne “true” si un dígito está presente en un número o “false” en caso contrario.
Ejemplos: el 2 está presente en 12345, pero el 7 no. el 0 está presente en 0, pero el 1 no.

Planteo: Dig presente en N
Caso base: si N tiene un único dígito ($N \div 10 = 0$) entonces si $N = \text{Dig}$ está presente, sino no lo está
Caso general: si N tiene más de un dígito, entonces Dig está presente si Dig es el último dígito de N o si está presente en N sin su último dígito.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

Problema propuesto: elemento presente

Escriba un planteo recursivo y luego una función (que respete ese planteo) que retorne “true” si un elemento está en un archivo o “false” en caso contrario.
Ejemplos: el 2 está presente en 1 2 3 4 5, pero el 7 no. Ningún elemento está presente en el archivo vacío.

Planteo: Ele está presente en archivo
Caso base: si el archivo está vacío, entonces retornar false
Caso general: si el archivo no está vacío, entonces retornar TRUE si Ele es igual al primer elemento, o ver si está en el archivo sin su primero elemento.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
“Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.

Problema propuesto: factorial

- La función factorial (!) puede definirse recursivamente de la siguiente manera:

$$N! \begin{cases} 0! = 1 \\ N! = N * (N-1)! \end{cases}$$

- Observe que tiene un **caso base** y otro caso que se define con una **instancia más simple** de sí mismo.

Planteo Recursivo para Factorial de N

Caso Base: si $N = 0$ Factorial de N es 1

Caso General:

si $N > 0$ entonces Factorial de N es $N * \text{Factorial de } N-1$

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

13

```

program Factorial-Recursivo;
var num, fact: integer;

Function Factorial (N:integer):integer;
var aux,nuevoN: integer;
begin
  IF N=0 THEN Factorial:=1
  ELSE begin
    nuevoN:=N-1;
    aux:= Factorial(nuevoN);
    Factorial:= N * aux;
  end;
end;

begin
  writeln(' Ingrese un número'); Readl(Num);
  fact:=factorial(Num);
  writeln(' Factorial de ', Num,' es ', Fact );
end.

```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

14

Otras dos formas de implementar la función

```

Function Factorial (N:integer):integer;
var aux: integer;
begin
  IF N=0
  THEN aux:=1
  ELSE aux:= N* Factorial(N-1);
  Factorial:= aux;
end;

```

```

Function Factorial (N:integer):integer;
begin
  IF N=0
  THEN Factorial :=1
  ELSE Factorial := N* Factorial(N-1);
end;

```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

15

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:

“Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.