




Resolución de Problemas y Algoritmos

Clase 18:
Resolución de problemas utilizando recursión



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

Recursión: 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 (aquí ver detalles de implementación, por ejemplo parámetros).
5. Realizar la **traza** de la primitiva en Pascal.

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

Problema propuesto

Escriba un planteo recursivo y luego un procedimiento que respete ese planteo, que dado un archivo de texto genere un nuevo archivo con los elementos en orden inverso.

Ejemplos:

Hola, nos vemos a las 14hs.
.hs41 sal a somev son ,aloh

Son \$2
2\$ noS

El inverso de un archivo vacío es un archivo vacío.

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

Problema propuesto

Escriba un planteo recursivo y luego un procedimiento que respete ese planteo, que dado un archivo de texto genere un nuevo archivo con los elementos en orden inverso.

Planteo: Archivo invertido de T

- Caso base: si T está vacío, entonces el archivo invertido de T es el archivo vacío.
- Caso general : si T tiene elementos entonces el archivo invertido de T es el archivo invertido de T sin su primer elemento, seguido del primer elemento de T.

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

```

Procedure invertir( var F1,F2: Text);
{Retorna en F2 el contenido del archivo F1 en forma invertida}
Procedure invertirRec( var Ori, Inv: Text);
{Retorna en Inv el contenido del archivo Ori en forma invertida}
var elemento: char;
begin
  IF EOF(Ori) THEN rewrite(Inv)
  ELSE begin
    read(Ori,elemento);
    invertirRec(Ori,Inv);
    write(Inv,elemento);
  end; {else}
end;
Begin
  reset(F1);
  invertirRec(F1,F2);
  close(F1); close (F2);
end
    
```

En esta versión, el archivo vacío es generado en el caso base.

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

```

Procedure invertir( var F1,F2: Text);
{Retorna en F2 el contenido del archivo F1 en forma invertida}
Procedure invertirRec( var Ori,Inv: Text);
{Retorna en Inv el contenido del archivo Ori en forma invertida}
var elemento: char;
begin
  IF not EOF(Ori) THEN
    begin read(Ori,elemento);
      invertirRec(Ori,Inv);
      write(Inv,elemento);
    end;
  end;
Begin
  Reset(F1); rewrite(F2);
  invertirRec(F1,F2);
  close(F1); close (F2);
end
    
```

En esta versión, como el archivo vacío es generado afuera el caso base no requiere de ninguna acción

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) 2014

```

program Reflexion4; {El objetivo de este programa es hacer una traza y
reflexionar sobre el pasaje de parámetros por referencia.}
var v1,v2,v3,v4:integer; {quedó un poco compacto para que entre en una "slide"}
procedure P3 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P3 con ', R:9, N:9);
local:= 3; N:= local+N; R:=N; C:=0; X:=R;
writeln('Salgo de P3 con ', local, R:9, C:9, X:9, N:9); end;
procedure P2 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P2 con ', R:9, N:9);
local:= 2; P3 (local,C,X,N+1); R:=local+N; C:=C+1;
writeln('Salgo de P2 con ', local, R:9, C:9, X:9, N:9); end;
procedure P1 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P1 con ', R:9, N:9);
local:= 1; P2 (local,C,X,N+1); R:=local+N; C:=C+1;
writeln('Salgo de P1 con ', local, R:9, C:9, X:9, N:9); end;
begin v1:=5; v4:=1; P1(v1,v2,v3,v4); write('finalizo con', v1,v2,v3,v4); end.
    
```

Tarea: Primero haga una traza en papel (bien prolija) y luego ejecute en su computadora para comparar. (Puede agregar mas "writeln's.")

Preguntas (¿teóricas?) para reflexionar

Las siguientes preguntas son sobre el programa "reflexion4",

(antes de responderlas tiene que hacer la traza)

- (fácil): ¿cuáles son parámetros por referencia?
- La variable v2 no tiene valor al ser usada en el parámetro efectivo de la llamada a P1, ¿es un error de programación?
- La variable v1 si tiene valor ¿es un error? ¿es mejor?
- ¿qué ocurriría si en P2 no se hiciera C:=0?

Parte de la traza

Estado de la traza antes de llamar a P1

reflexion4	
v1	5
v2	
v3	
v4	1

En esta página y las siguientes se muestran algunas partes de la traza del programa reflexion4. Sugerencia: realice su propia traza completa y compare.

Estado de la traza luego de ejecutar "local:= 3" en P3

reflexion4	P1	P2	P1
v1	5	5	5
v2			
v3			
v4	1	1	1
	local 1	local 2	local 3
	R	R	R
	C	C	C
	X	X	X
	N 1	N 2	N 3

Parte de la traza

Luego de ejecutar "X:= R" en P3: observe los cambios en v2 y v3, (y local de P2)

reflexion4	P1	P2	P1
v1	5	5	5
v2	0	0	0
v3	6	6	6
v4	1	1	1
	local 1	local 6	local 3
	R	R	R
	C	C	C
	X	X	X
	N 1	N 2	N 6

Luego de ejecutar "C:=C+1;" en P2: observe los cambios en v2 y v3

reflexion4	P1	P2
v1	5	5
v2	1	1
v3	6	6
v4	1	1
	local 8	local 6
	R	R
	C	C
	X	X
	N 1	N 2

Parte de la traza

Luego de ejecutar "C:=C+1;" en P1: observe los cambios en v2 y v3

reflexion4	P1
v1	9
v2	2
v3	6
v4	1
	local 8
	R
	C
	X
	N 1

Estado final de las variables globales.

reflexion4	
v1	9
v2	2
v3	6
v4	1

Problema propuesto: cantidad de elementos

Escriba un planteo recursivo y luego un procedimiento que respete ese planteo para contar la cantidad de apariciones de un elemento de un archivo.

Ejemplo, el 3 está 2 veces en el archivo F: 4 3 3 (como fue hecho en el pizarrón, realice la traza con el ejemplo anterior y compare con al traza de reflexion4)

Planteo: Cantidad de apariciones de E en F

Caso base: Si F está vacío,
la cantidad de apariciones de E en F es 0.

Caso general: La cantidad de apariciones de E en F, es la cantidad de apariciones de E en F sin su primer elemento, más uno si el primer elemento de F es E.

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) 2014

```


program prueba1;
type Telemento = integer; Tarchi = file of Telemento;
var A: Tarchi; cantidad: integer; E: Telemento;

Procedure contar (E: Telemento; var F: Tarchi; var cant: integer);
{cuenta las apariciones de E en un archivo F}
var aux: telemento;
begin
  if EOF(F) then cant:=0 {caso base}
  else begin read(F,aux); {caso recursivo}
            contar(E, F, cant);
            if aux = E then cant:= cant +1; end;
end;

Procedure leer elemento(var E: Telemento); {... completar...}
begin
assign (A, 'el-archivo'); leer_elemento(E); {primitiva que valida}
reset(A); contar(E, A, cantidad); close(A);
writeln('cantidad de apariciones: ',cantidad);
end.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

Fibonacci (1170 -1250)



Leonardo de Pisa, matemático italiano. El apodo de su padre era *Bonacci* (bien intencionado) y él recibió el apodo *Fibonacci: filius* (hijo de) Bonacci. Su padre era comerciante, y Fibonacci de joven vivió en África, donde estudió con los matemáticos árabes más destacados de ese tiempo. Allí aprendió el sistema de numeración árabe (decimal). Consciente de la superioridad de este sistema comparado con el romano, en 1202, a los 32 años de edad, publicó lo que había aprendido en el **Liber Abaci** (libro del ábaco o libro de los cálculos), mediante el cual se introdujo en Europa el sistema decimal que reemplazaría al romano.

http://es.wikipedia.org/wiki/Leonardo_de_Pisa

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

Sistema de numeración decimal (base 10)

- El sistema de numeración decimal se considera uno de los avances más significativos de las matemáticas.
- La mayoría de los historiadores coinciden en afirmar que tuvo su origen en la India (Tamil) en 300 aC (pero también se especula que tuviera sus orígenes en China).
- Este sistema de numeración llegó a **Oriente Medio** hacia el año 670. **al-Jwarizmi** escribió el libro "Acercas de los cálculos con los números de la India" cerca de el año 825.
- En Europa se utilizaban los números Romanos, pero **Fibonacci**, que había estudiado en **Bugia** (en la actual **Argelia**), contribuyó a la difusión por **Europa** del sistema arábigo con su libro **Liber Abaci**, publicado en 1202.

http://es.wikipedia.org/wiki/Números_arabigos

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

Evolución de los símbolos de los dígitos

1200	Europeo	0	1	2	3	4	5	6	7	8	9
	Arábico-Índico	•	١	٢	٣	٤	٥	٦	٧	٨	٩
670	Arábico-Índico Oriental (Persa y Urdu)	•	١	٢	٣	٤	٥	٦	٧	٨	٩
	Devanagari (Hindi)	०	१	२	३	४	५	६	७	८	९
300aC	Tamil (India)		௧	௨	௩	௪	௫	௬	௭	௮	௯

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 16

Sucesión de Fibonacci

La **sucesión de Fibonacci** es una sucesión infinita de números naturales: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Que inicia con 0 y 1, y a partir de ahí cada elemento es la suma de los dos anteriores. A cada elemento de esta sucesión se le llama **número de Fibonacci**.

La sucesión fue descrita por Fibonacci, en su libro *Liber Abaci*, como la solución a un problema de la cría de conejos. http://es.wikipedia.org/wiki/Sucesion_de_Fibonacci

Antes de que Fibonacci escribiera su trabajo, la sucesión de los números de Fibonacci había sido descubierta por matemáticos indios tales como Gopala (antes de 1135) y Hemachandra (1150),

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

Sucesión de Fibonacci: definición recursiva

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Los números de Fibonacci $f_0, f_1, f_2, f_3, f_4, \dots$ pueden definirse recursivamente como:

- $f_0 = 0$
- $f_1 = 1$
- $f_n = f_{n-1} + f_{n-2}$ para $n > 1$

Ejemplo:

$$f_4 = f_3 + f_2 = (f_2 + f_1) + (f_1 + f_0) = ((f_1 + f_0) + f_1) + (f_1 + f_0) = 3$$

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 18

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) 2014