

## RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

### CLASE 21

#### Resolución de problemas utilizando recursión

Luciano H. Tamargo  
<http://cs.uns.edu.ar/~lt>  
 Depto. de Ciencias e Ingeniería de la Computación  
 Universidad Nacional del Sur, Bahía Blanca  
 2016

```
0 1 1 0 0
1 0 0 1 1
1 0 1 1 0
0 1 1 1 0
0 1 1 0 0
1 0 0 1 1
1 0 1 1 0
1 1 1 1 0
0 0 1 1
1 1
0
```

### METODOLOGÍA (Y TÉCNICAS) PROPUESTAS

- Al abordar el problema:
  - Utilizar la técnica de **división en sub-problemas** si fuera necesario.
  - Hacer ejemplos significativos particulares que ayuden a entender el problema (o cada sub-problema): esta técnica también se denomina **particularización**.
  - Generalizar las ideas de los ejemplos particulares para cubrir todos los casos: esta técnica también se denomina **generalización**.
- Buscar si existe alguna **analogía** con otros problemas ya resueltos para aprovechar la experiencia anterior.
- Si se quiere una solución recursiva: realizar un **planteo recursivo** en el cual se distinga el "caso base", y el "caso general" (donde se define en términos de sí mismo pero para una instancia más simple/reducida/menor).
- Verificar que el planteo recursivo (o el algoritmo propuesto) sea correcto (usar los ejemplos del punto 1).
- Determinar si se realizará una **función** o un **procedimiento recursivo**, e implementarlo en Pascal.
- Realizar la **traza** de cada primitiva en Pascal.

Resolución de Problemas y Algoritmos - 2016

2

### PROBLEMA PROPUESTO

**Problema propuesto:** Escriba un planteo recursivo y luego un procedimiento recursivo que respete ese planteo, que dado un archivo de texto llamado "original.txt" genere un nuevo archivo (llamado "invertido.txt") con los elementos del original pero en orden inverso.

#### Tres ejemplos:

Hola, nos vemos a las 14hs.  
 .sh41 sal a somev son ,aloH

Abc  
 cbA

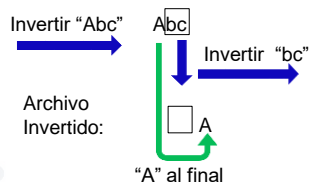
*El inverso de un archivo vacío es un archivo vacío.*

Resolución de Problemas y Algoritmos - 2016

3

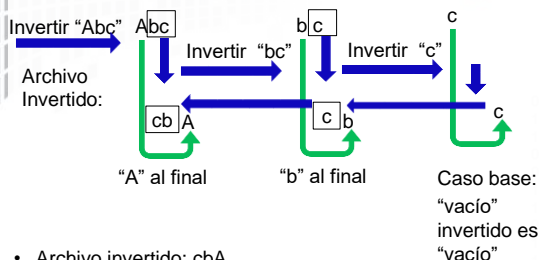
### PROBLEMA PROPUESTO

- Por ejemplo:** observe que al invertir el archivo con el contenido "Abc", el primer elemento ("A") debe quedar al final del archivo invertido, y el resto de la secuencia ("bc"), debe quedar al principio del archivo invertido, pero a su vez en orden inverso (hay que invertir "bc").



### PROBLEMA PROPUESTO

- Archivo original: Abc



- Archivo invertido: cbA



### PROBLEMA PROPUESTO

**Problema propuesto:** Escriba un planteo recursivo y luego un procedimiento recursivo que respete ese planteo, que dado un archivo de texto llamado "original.txt" genere un nuevo archivo (llamado "invertido.txt") con los elementos del original pero en orden inverso.

- Por ejemplo:** observe que al invertir el archivo con el contenido "Abc", el primer elemento ("A") debe quedar al final del archivo invertido, y el resto de la secuencia ("bc"), debe quedar al principio del archivo invertido, pero a su vez en orden inverso (hay que invertir "bc").

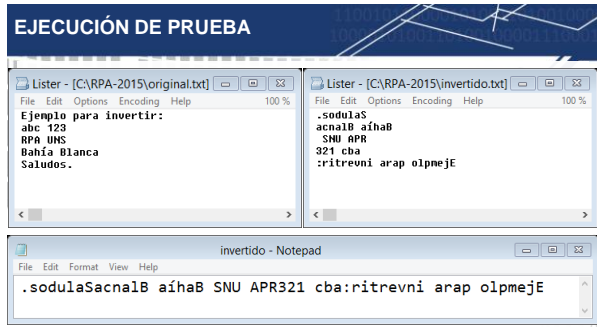
#### Planteo recursivo: Archivo invertido de T

- 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.



```
PROGRAM prueba;
VAR F1, F2: Text;

PROCEDURE invertirArchivo(VAR Ori, Inv: Text);
{Retorna en Inv el contenido del archivo Ori en
orden inverso}
VAR ele: char;
BEGIN
IF EOF(Ori) THEN {caso base}
write(Inv)
ELSE
BEGIN {caso general}
read(Ori, ele);
invertirArchivo(Ori, Inv);
write(Inv, ele);
END;
END;
BEGIN
assign(F1, 'original.txt'); assign(F2, 'invertido.txt');
reset(F1); invertirArchivo(F1, F2);
close(F1); close(F2);
END.
```



¿Por qué "Notepad" lo muestra como una sola línea?  
 ¿Puede cambiar la implementación para que esto no ocurra?

## PROBLEMA PROPUESTO

**Problema propuesto:** Escriba un programa que permita ingresar por teclado una secuencia de caracteres terminada en un punto (por ejemplo: "hola que tal.") y que la muestre por pantalla en orden inverso ("lat euq aloh").

- **Ejemplo:**

```
Ingrese una cadena terminada en punto:
abc 123.
Invertida queda así:
321 cba
```

**Planteo recursivo: Mostrar Invertida la secuencia S**

**caso base:** si la secuencia **S** es solamente un ".", entonces el orden inverso es la secuencia vacía.

**caso general:** si **S** tiene más de un elemento, entonces: **Mostrar Invertida la secuencia S sin su primer elemento**, y luego mostrar el primer elemento de **S**.

```
PROCEDURE MostrarInvertida; {observe que no hay parámetros}
VAR caract: char;
BEGIN
read(caract); {leo el primer elemento de la secuencia}
IF caract = '.' THEN
write('Invertida queda así: ') {caso base}
ELSE
BEGIN {caso general}
MostrarInvertida; {llamada recursiva}
write(caract); {imprime el primer elemento}
END; {fin del caso general}
END;
```

**Planteo recursivo: Mostrar Invertida la secuencia S**

**caso base:** si la secuencia **S** es solamente un ".", entonces el orden inverso es la secuencia vacía.

**caso general:** si **S** tiene más de un elemento, entonces: **Mostrar Invertida la secuencia S sin su primer elemento**, y luego mostrar el primer elemento de **S**.

```
PROGRAM prueba_buffer; {Programa de prueba}

PROCEDURE MostrarInvertida; {observe que no hay parámetros}
VAR caract: char;
BEGIN
read(caract); {leo el primer elemento de la secuencia}
IF caract = '.' THEN
write('Invertida queda así: ') {caso base}
ELSE
BEGIN {caso general}
MostrarInvertida; {llamada recursiva}
write(caract); {imprime el primer elemento}
END; {fin del caso general}
END;

BEGIN
write('Ingrese una cadena terminada en punto:');
MostrarInvertida;
END.
```