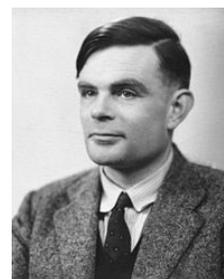


Resolución de Problemas y Algoritmos

Clase 20 Resolución de problemas utilizando recursión



Alan M. Turing



Dr. Diego R. García



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



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.

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").

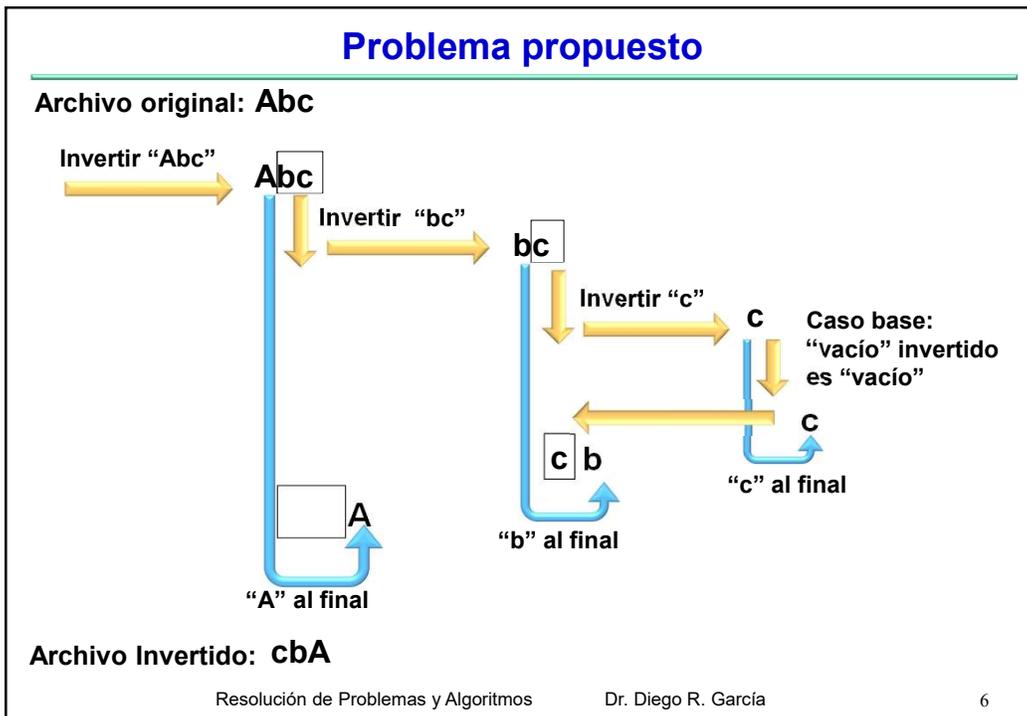
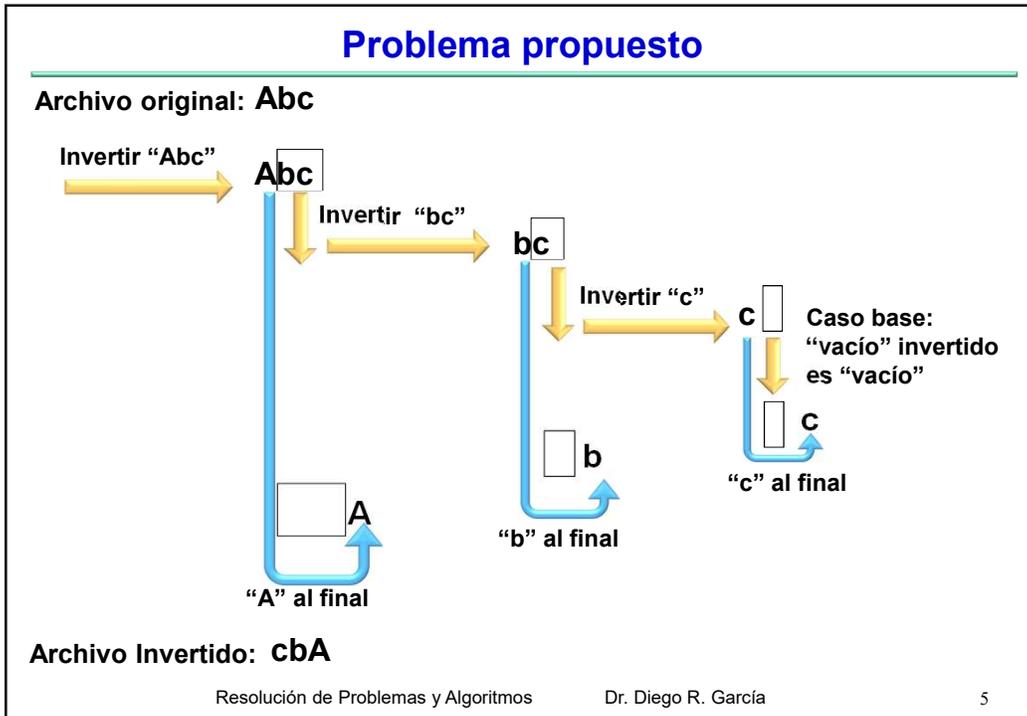
Resolución de Problemas y Algoritmos Dr. Diego R. García 3

Problema propuesto

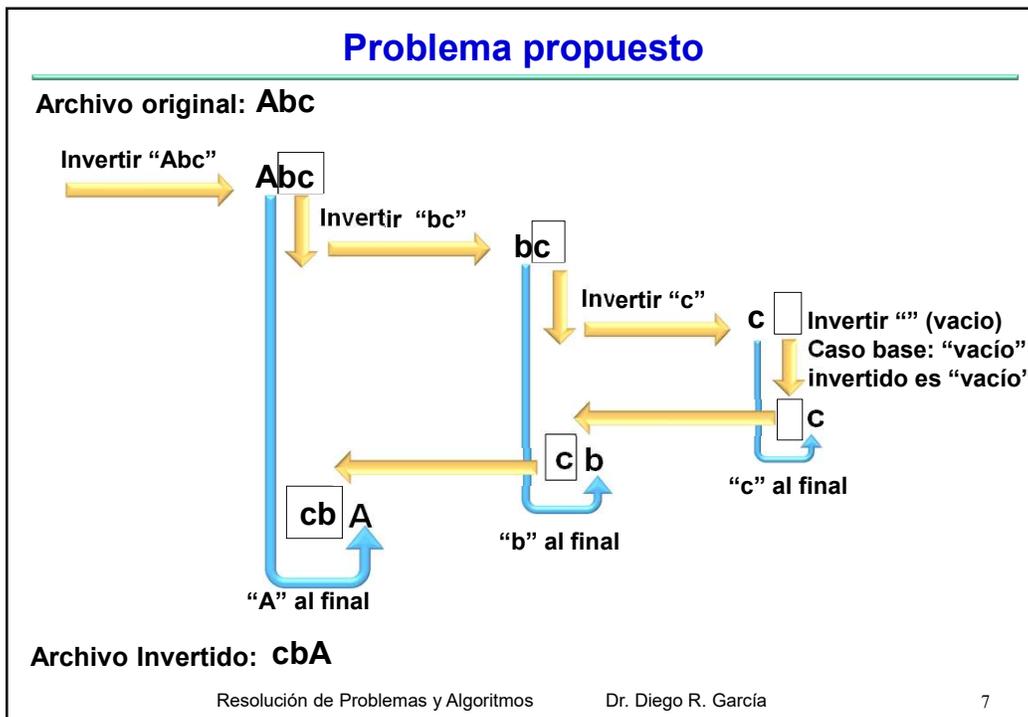
Archivo original: Abc

Resolución de Problemas y Algoritmos Dr. Diego R. García 4

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". Diego R. García. Universidad Nacional del Sur. (c) 07/11/2019



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". Diego R. García. Universidad Nacional del Sur. (c) 07/11/2019



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.**

Resolución de Problemas y Algoritmos Dr. Diego R. García 8

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". Diego R. García. Universidad Nacional del Sur. (c) 07/11/2019

```
Program prueba;  
var F1,F2: Text;  
Procedure invertirArchivo( var Ori, Inv: Text);  
{Retorna en Inv el contenido del archivo Ori en orden inverso}  
var elemento: char;  
begin  
  IF EOF(Ori) THEN rewrite(Inv) {caso base}  
  ELSE begin {caso general}  
    read(Ori,elemento);  
    invertirArchivo(Ori,Inv);  
    write(Inv,elemento);  
  end;  
end;  
Begin  
assign(F1,'original.txt'); assign(F2,'invertido.txt');  
reset(F1);  
invertirArchivo(F1,F2);  
close(F1); close (F2);  
end
```

Resolución de Problemas y Algoritmos

Dr. Diego R. García

9

```
Program prueba2;  
var F1,F2: Text;  
Procedure invertirArchivo( var Ori, Inv: Text);  
{Retorna en Inv el contenido del archivo Ori en orden inverso}  
var elemento: char;  
begin  
  IF NOT EOF(Ori)  
  THEN begin {caso general}  
    read(Ori,elemento);  
    invertirArchivo(Ori,Inv);  
    write(Inv,elemento);  
  end;  
end;  
Begin  
assign(F1,'original.txt'); assign(F2,'invertido.txt');  
reset(F1); rewrite(F2);  
invertirArchivo(F1,F2);  
close(F1); close (F2);  
end
```

Resolución de Problemas y Algoritmos

Dr. Diego R. García

10

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”. Diego R. García. Universidad Nacional del Sur. (c) 07/11/2019

Ejecución de prueba

File Edit Options Encoding Help 100 %

Ejemplo para invertir:
 abc 123
 RPA UNS
 Bahía Blanca
 Saludos.

File Edit Options Encoding Help 100 %

```
.sodulaS
acna1B aÍhaB
SNU APR
321 cba
:ritrevni arap olpmejE
```

invertido - Notepad

File Edit Format View Help

```
.sodulaSacna1B aÍhaB SNU APR321 cba:ritrevni arap olpmejE
```

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

Resolución de Problemas y Algoritmos
Dr. Diego R. García
11

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.

Resolución de Problemas y Algoritmos
Dr. Diego R. 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”. Diego R. García. Universidad Nacional del Sur. (c) 07/11/2019

procedimiento recursivo: invertir buffer

```

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
  writeln(' Ingrese una cadena terminada en punto: ');
  MostrarInvertida;
end.

```

Tarea propuesta

A continuación hay varios 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) Realice la traza.
- 4) Pase en la máquina y vea la ejecución
- 5) Consulte sus dudas

Problemas propuestos

<1> Escriba un planteo recursivo y luego una primitiva que respete ese planteo para **contar la cantidad de letras ingresadas por teclado, de una secuencia finalizada en un punto.**

<2> Escriba un planteo recursivo y luego una primitiva en Pascal (que respete ese planteo) que **indique cuando un dígito está presente en un número entero.**

<3> Escriba un planteo recursivo y luego una primitiva en Pascal (que respete ese planteo) que **indique si un elemento está presente o no en un archivo.**

Problema propuesto 1: 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. (3 elem.) a. (1 elem) . (0 elem.)

Planteo: Cantidad de letras de una secuencia

Caso base: si la secuencia es solamente un “.” entonces la cantidad de letras es 0

Caso general: si la secuencia tiene más de un elemento, entonces la cantidad es 1 + la cantidad de letras de la secuencia sin su primero elemento.

Problema propuesto 2: dígito presente

Escriba un planteo recursivo y luego una primitiva en Pascal (que respete ese planteo) que indique cuando un dígito está presente en un número entero.

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

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.

Problema propuesto 3: elemento presente

Escriba un planteo recursivo y luego una primitiva en Pascal (que respete ese planteo) que **indique si un elemento está presente o no en un archivo**.

Ejemplos: el 2 está presente en 1 2 3 4 5, pero el 7 no.

Ningún elemento está presente en el archivo vacío.

Planteo: Elem está presente en archivo

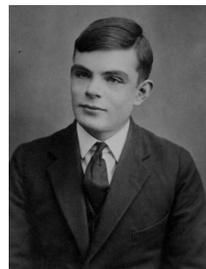
Caso base: si el archivo está vacío,
entonces no está el elemento

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

Información adicional

Alan Mathison Turing, (23/6/1912 - 7/6/1954)

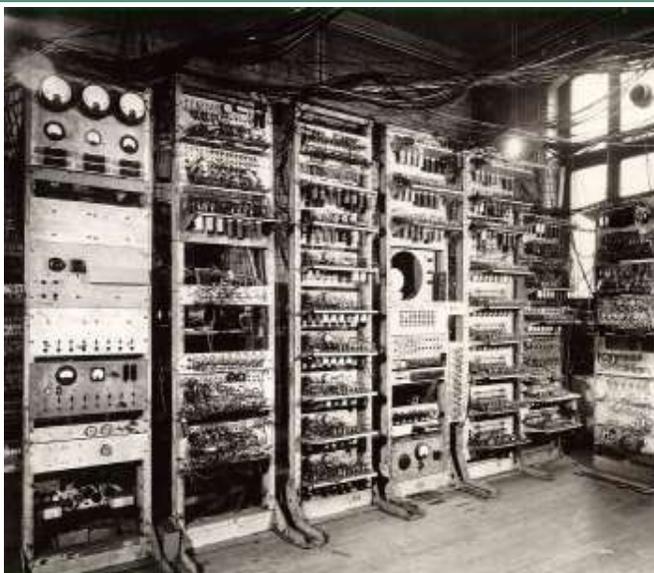
- Matemático, científico de computación, lógico, criptógrafo y filósofo británico.
- En 1936 publicó la definición teórica de su “*Universal computing machine*”.
- En 1940 junto a otros desarrolló la idea de que en la misma memoria de un computadora estén programa y datos.
- En 1948 fue nombrado director of Computing Machine Laboratory at the University of Manchester donde se estaba construyendo “Manchester Mark 1” (ver foto siguiente)



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

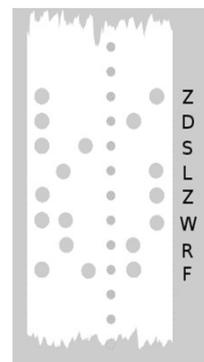
Primeras computadoras: Mark 1

El desarrollo de Mark 1 comenzó en 1948 y se basó en la arquitectura de programa almacenado. Estuvo completamente operable para octubre de 1949.



Alan Turing y Manchester Mark 1

- Manchester Mark 1 disponía de una RAM construida con un tubo de rayos catódicos ([Williams–Kilburn tube](#))
- Mark 1 no tenía sistema operativo, ni lenguaje ensamblador, los programas debían ser escritos en binario.
- Alan Turing desarrolló para Mark 1 un esquema de codificación que permitía que programas y datos sean escritos y leídos de una cinta de papel.



http://en.wikipedia.org/wiki/Manchester_Mark_1

Alan M. Turing, y la Inteligencia Artificial

En 1950 contribuyó de forma particular e incluso provocativa a la Inteligencia Artificial con su artículo **Computing Machinery and Intelligence**, el cual comienza así: I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think"....

Luego propone lo siguiente:

"En lugar de intentar dar estas definiciones, reemplazaré la pregunta por otra, que está estrechamente relacionada pero está expresada en términos relativamente sin ambigüedad. La nueva versión del problema puede describirse en términos de un juego al cual llamaremos "juego de la imitación". (el texto completo está en [aquí](#))



Juego de la imitación (Imitation Game) Turing 1950

Se juega con tres personas. Un hombre (A), una mujer (B) y (C) un interrogador (de cualquier sexo). El interrogador se encuentra separado de las otras dos (de manera de no poder verlas u oírlas) a las cuales conoce como X e Y.

El objetivo del juego es que C debe determinar cual de los otros dos es A y cual es B. Durante el juego, C puede hacer todo tipo de preguntas a X e Y, (escritas usando algún medio de comunicación). En el juego, el objetivo de A es engañar a C intentando que piense que es la mujer y el de B ayudar a C. Al final del juego C debe indicar si X es A o X es B.

Ahora hacemos la pregunta, “¿qué pasaría si una máquina toma el lugar de A en el juego?” ¿El interrogador decidiría equivocadamente tan a menudo como cuando en el juego había dos personas?. Esta pregunta reemplaza a la original, “¿Pueden las máquinas pensar?”.

Test de Turing

El juego de la imitación dio paso a un test (llamado Turing Test) en el cual una persona P se comunica (sin verlos) con dos entidades A y B, donde una de las dos es una persona y la otra es una máquina. Durante el test, P puede hacer todo tipo de preguntas a B y A, e intentar descubrir cual de las dos es la máquina. Si P no puede distinguir quien de los dos es la máquina, entonces la máquina estará demostrando un comportamiento similar al de una persona.

En 1990 se inició un concurso, el Premio Loebner. El premio está dotado con 100.000 dólares para el programa que pase el test, y un premio de consolación para el mejor programa anual.

[http://es.wikipedia.org/wiki/Test de Turing](http://es.wikipedia.org/wiki/Test_de_Turing)

[http://en.wikipedia.org/wiki/Loebner Prize](http://en.wikipedia.org/wiki/Loebner_Prize)

Captcha (o CAPTCHA)

Completely Automated Public Turing test to tell Computers and Humans Apart (Prueba de Turing pública y automática para diferenciar máquinas de humanos). Este test es controlado por una máquina (con un algoritmo público), en lugar de por un humano como en la Prueba de Turing.



Ejemplo: Test CAPTCHA para la secuencia «smwm» que dificulta el reconocimiento OCR por parte de los spambots.

Se trata de una prueba para determinar cuándo el usuario es o no humano. Consiste en que el usuario introduzca correctamente un conjunto de caracteres que se muestran en una imagen distorsionada que aparece en pantalla. Se supone que una máquina (aún) no es capaz de comprender e introducir la secuencia de forma correcta por lo que solamente un humano podría hacerlo. <http://es.wikipedia.org/wiki/Captcha>

