




Redictado Taller de programación 2020

CLASE 2

Parte I – Conceptos teóricos de la Recursión



```
Program HolaMundo;  
Begin  
  writeln('Hola mundo');  
end.
```



Temas de la clase

- Recursión. Concepto. Motivación
- Ejemplo de recursión
- ¿Cómo funciona la recursión?
- Características de un algoritmo recursivo
- Ejercitación



Recursión

La recursión es una metodología para resolver problemas.

Permite resolver un problema **P** por resolución de instancias más pequeñas **P₁**, **P₂**, ..., **P_n** del mismo problema.

El problema **P_i** es de la misma naturaleza que el problema original, pero en algún sentido es más simple.



Veamos un ejemplo ...



Recursión

Problema:
Buscar una
palabra en un
diccionario que
tiene 2000
páginas.

Solución:
Paso 1)
Abrir el diccionario
a la mitad



Sofi



Recursión

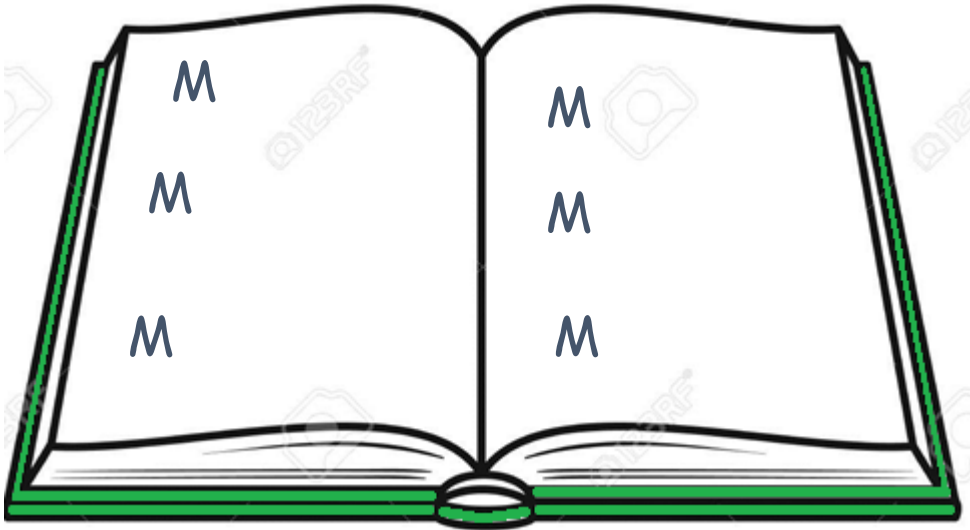
Problema:

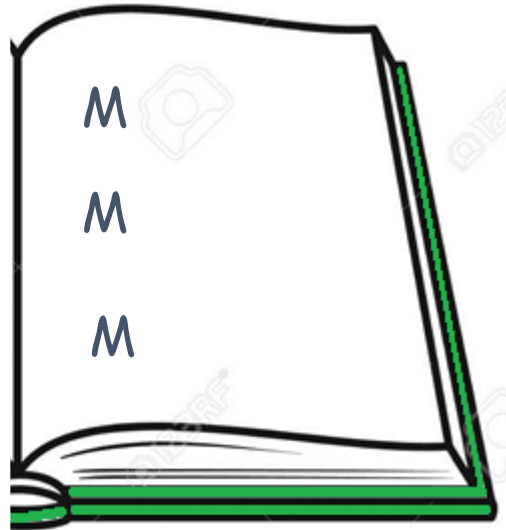
Buscar una palabra en un diccionario que tiene 2000 páginas.

Solución:

Paso 2)

Quedarse con la mitad donde está la palabra





Recursión

Problema:

Buscar una palabra en un diccionario que tiene 1000 páginas.

Solución:

Paso 1)
Abrir el diccionario a la mitad



Sofi



Recursión

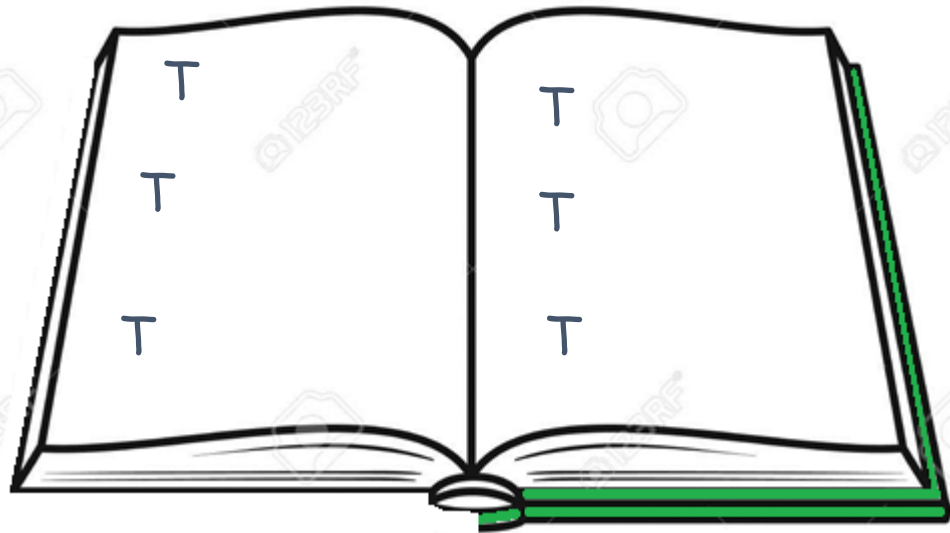
Problema:

Buscar una palabra en un diccionario que tiene 1000 páginas.

Solución:

Paso 2)

Quedarse con la mitad donde está la palabra





Sofi



Recursión

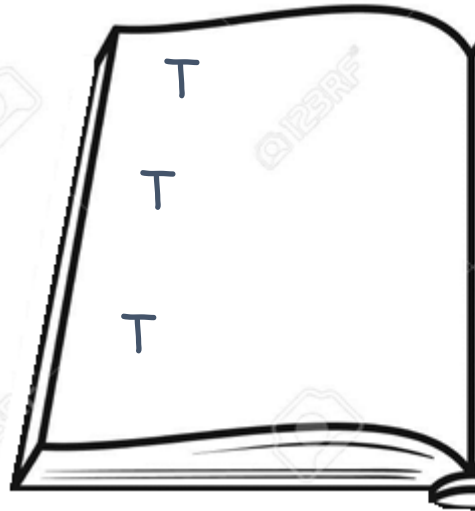
Problema:

Buscar una palabra en un diccionario que tiene 500 páginas.

Solución:

Paso 1

Paso 2



Solución recursiva



Sofi



Problema:

Buscar una palabra en un diccionario.

Caso recursivo:

- ❖ El problema es siempre el mismo, pero en cierto sentido es cada vez más pequeño
- ❖ La tarea a realizar es siempre la misma y en cierto sentido consiste en reducir el espacio donde hacer la búsqueda

Solución:

Paso 1) Abrir el diccionario a la mitad.

Paso 2) Quedarse con la mitad donde está la palabra.



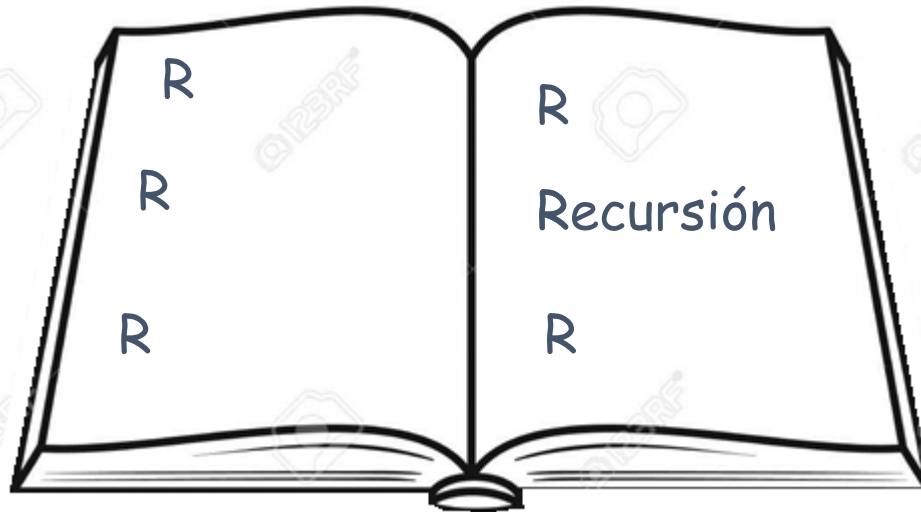
Sofi



Recursión

¿Cuándo termino?

a) En algún momento abrirás el diccionario a la mitad y encontraras la palabra.



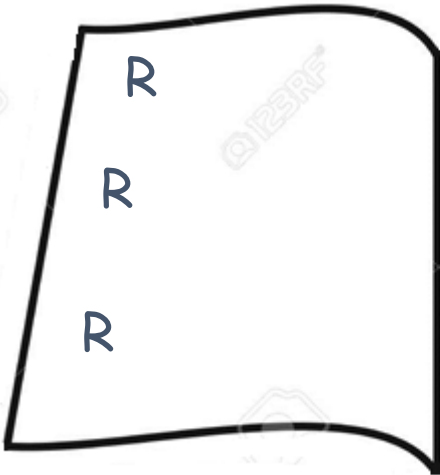


Sofi

Recursión

¿Cuándo termino?

b) Si ya no puedes seguir partiendo a la mitad el diccionario, la palabra buscada no está.



Solución recursiva



Sofi

Caso base:

- ❖ Es una instancia del problema donde no se puede seguir dividiendo el problema
- ❖ El caso base termina con la recursión

¿Cuándo termino?

- a) Se encuentra la palabra.
- b) La palabra buscada no existe.





Recursión

Características de un algoritmo recursivo

Una **solución recursiva** resuelve un problema por resolución de instancias más pequeñas del mismo problema.

Un algoritmo recursivo involucra:

- al menos una condición de terminación (implícita / explícita) (Caso base)
- al menos una *autoinvocación* (Caso recursivo). Se debe garantizar que en un número finito de *autoinvocaciones* se alcanza la condición de terminación

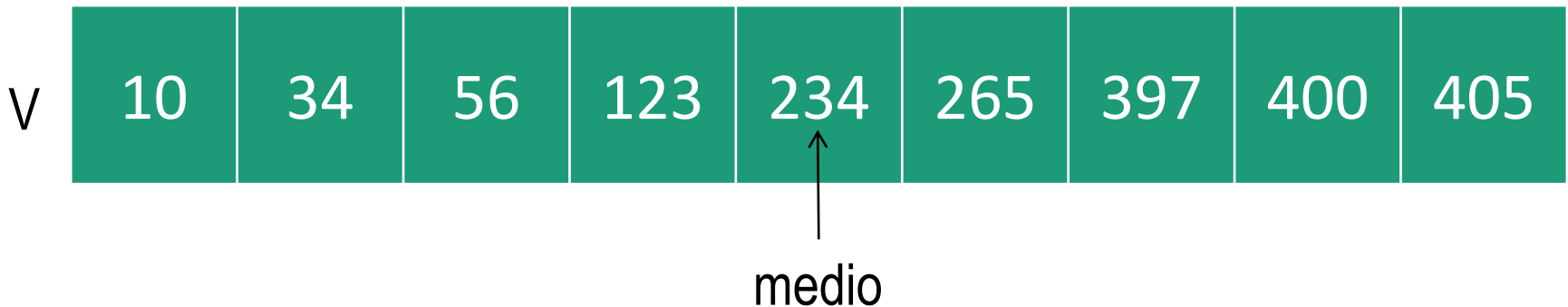


Recursión

Ejemplo: Búsqueda dicotómica en un vector



Buscar el valor 56 en el vector



¿Cómo es 56 con respecto a $v[\text{medio}]$?

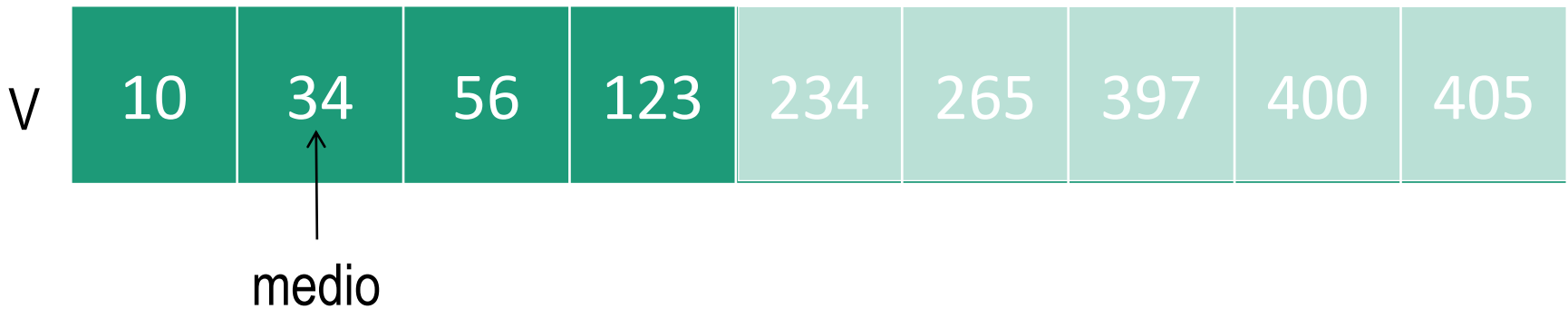
1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior

Recursión



Ejemplo: Búsqueda dicotómica en un vector

Buscar el valor 56 en el vector



¿Cómo es 56 con respecto a $v[\text{medio}]$?

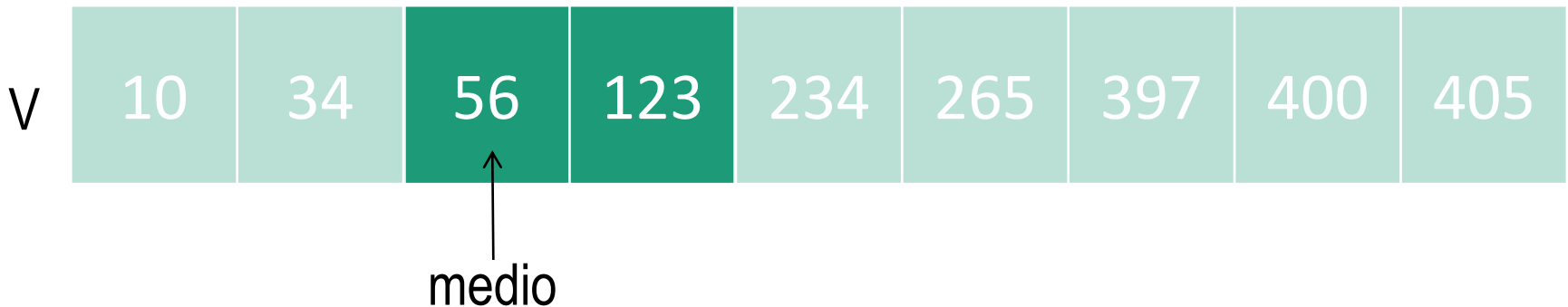
1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior

Recursión



Ejemplo: Búsqueda dicotómica en un vector

Buscar el valor 56 en el vector



¿Cómo es 56 con respecto a $v[\text{medio}]$?

1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior



Recursión

Ejemplo: Búsqueda dicotómica en un vector

Buscar el valor 56 en el vector

v	10	34	56	123	234	265	397	400	405
---	----	----	----	-----	-----	-----	-----	-----	-----

Observemos que :

1. La primera vez se trabaja con el vector completo para determinar el punto medio
2. La siguiente vez, el vector se reduce a la mitad
3. La siguiente vez, el vector se reduce a la mitad de la mitad

¿Cómo se calcula el medio?

¿Cómo se calcula la primera mitad?

¿Cómo se calcula la segunda mitad?



Recursión

Ejemplo: Búsqueda dicotómica en un vector

Buscar (vector, datoABuscar)

si el vector “no tiene elementos” entonces

No lo encontré y termino la búsqueda

sino

Determinar el punto medio del vector

Comparar **datoABuscar** con el contenido del punto medio

si coincide entonces

“Lo encontré”

sino

si **datoABuscar** < contenido del punto medio entonces

Buscar (1era mitad del vector, **datoABuscar**)

sino

Buscar (2da mitad del vector, **datoABuscar**)

3) Existen 2 casos que se resuelven de manera directa (casos base):

- a) Cuando el vector “no contiene elementos”
- b) Cuando encuentro el **datoABuscar**

2) En cada llamada, el tamaño del vector se reduce a la mitad.

1) El módulo realiza invocaciones a si mismo (Caso recursivo)

Recursión



Implementación de un Procedure o Function

Ejemplo: Búsqueda dicotómica en un vector

Buscar (vector, datoABuscar)

si el vector “no tiene elementos” entonces

 No lo encontré y termino la búsqueda

sino

 Determinar el punto medio del vector

 Comparar **datoABuscar** con el contenido del punto medio

 si coincide entonces

 “Lo encontré”

 sino

 si **datoABuscar** < contenido del punto medio entonces

Buscar (1era mitad del vector, **datoABuscar**)

 sino

Buscar (2da mitad del vector, **datoABuscar**)

Recursión



Implementación del
caso base

Ejemplo: Búsqueda dicotómica en un vector

Buscar (vector, datoABuscar)

si el vector “no tiene elementos” entonces

 No lo encontré y termino la búsqueda

sino

 Determinar el punto medio del vector

 Comparar **datoABuscar** con el contenido del punto medio

 si coincide entonces

 “Lo encontré”

sino

 si **datoABuscar** < contenido del punto medio entonces

Buscar (1era mitad del vector, datoABuscar)

 sino

Buscar (2da mitad del vector, datoABuscar)



Implementación del
caso recursivo
(autoinvocación)

Recursión

Ejemplo: Búsqueda dicotómica en un vector

Buscar (vector, datoABuscar)

si el vector “no tiene elementos” entonces

No lo encontré y termino la búsqueda

sino

Determinar el punto medio del vector

Comparar **datoABuscar** con el contenido del punto medio

si coincide entonces

“Lo encontré”

sino

si **datoABuscar** < contenido del punto medio entonces

Buscar (1era mitad del vector, **datoABuscar**)

sino

Buscar (2da mitad del vector, **datoABuscar**)

Recursión



Ejemplo: Potencia de un número

```
program CalculoDePotencia;  
var base, exponente, potencia, i: integer;  
begin  
  base := 2;  
  exponente := 3;  
  potencia := 1;  
  for i := 1 to exponente do  
    potencia := potencia * base;  
  writeln(potencia),  
  readln;  
end.
```

Planteo de solución recursiva. Tener en cuenta:

1. ¿Cómo defino el problema en términos de problemas más simples del mismo tipo?
2. ¿Cómo achico el problema en cada llamado recursivo?
3. ¿Qué instancia/s del problema son caso/s base?

Cálculo de 2^3 :

$$2^3 = 2 * 2^2 = 2 * \underbrace{2 * 2^1}_{2^2} = 2 * 2 * \underbrace{2 * 2^0}_{2^1} = 2 * 2 * 2 * \underbrace{1}_{2^0} = 8$$
A diagram illustrating the recursive calculation of 2^3. It shows the equation 2^3 = 2 * 2^2 = 2 * 2 * 2^1 = 2 * 2 * 2 * 2^0 = 2 * 2 * 2 * 1 = 8. Brackets are used to group the terms: a green bracket under 2 * 2^1 is labeled 2^2, a green bracket under 2 * 2^0 is labeled 2^1, and a green bracket under 1 is labeled 2^0. Two orange curved arrows are drawn: one from 2^2 to 2^1 and another from 2^1 to 2^0, indicating the recursive steps.

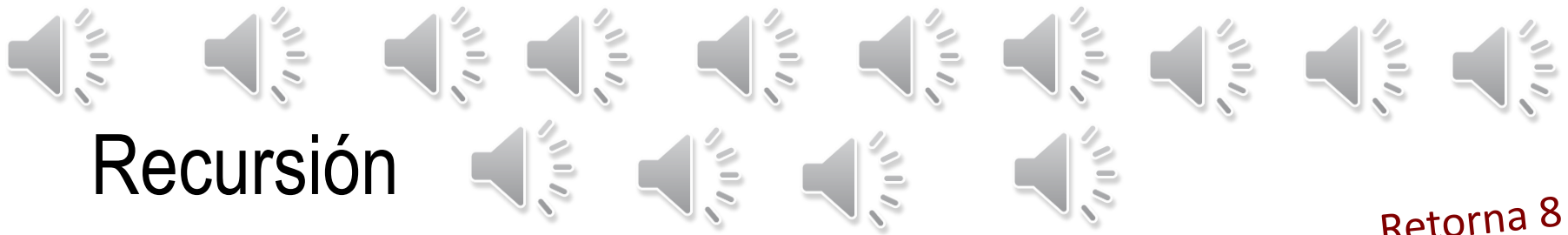


Recursión

Ejemplo: Potencia de un número

$$X^n = \begin{cases} 1 & \text{si } n = 0 \\ X * X^{n-1} & \text{si } n \geq 1 \end{cases}$$

```
Function potencia (x, n: integer): real;  
begin  
  if (n = 0) then  
    potencia := 1  
  else  
    potencia := x * potencia(x, n-1);  
  end;  
end;
```



Recursión

Ejemplo: Potencia de un número

```
program ejemplo;  
  
function potencia (x,n:integer): real;  
begin  
    if (n = 0) then  
        potencia := 1  
    else  
        potencia := x * potencia(x,n-1);  
    end;  
  
var  
    x,n:integer;  
begin  
    read (x,n);  
    write (potencia(x,n));  
end.
```

