



```
READ X,Y
```

```
IF X<Y
```

```
PRINT Y
```

```
ELSE
```

```
int
```

```
public
```

```
static
```

```
return
```

```
codio
```

Conceptos de Algoritmos Datos y Programas

Temas de la clase de hoy

1

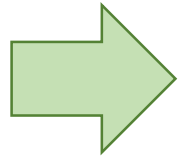
Corrección

2

Eficiencia

CORRECCIÓN de algoritmos

CORRECCIÓN



Un programa es correcto si se realiza de acuerdo a sus especificaciones.

TÉCNICAS

TESTING

DEBUGGING

WALKTHROUGHS

VERIFICACIÓN

CARACTERÍSTICAS

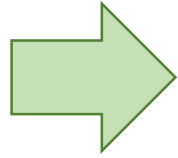
Cada prueba se realiza en un momento determinado.

Para comprobar que un programa es correcto, puede realizarse una o más pruebas.

Las pruebas deben repetirse hasta poder garantizar que el programa es correcto.

EFICIENCIA de algoritmos- Memoria

EFICIENCIA



La eficiencia de un programa se evalúa una vez que el programa es correcto. Está relacionado con los conceptos **MEMORIA** y **TIEMPO DE EJECUCIÓN**.

El usuario puede querer desarrollar un programa eficiente en cuanto a la memoria y/o tiempo de ejecución.

La **memoria total de un programa** = memoria estática + memoria dinámica.

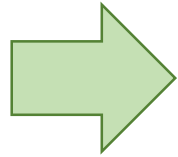
MEMORIA

Memoria estática: se calcula teniendo en cuenta, constantes + variables globales + variables locales del programa.

Memoria dinámica: se calcula teniendo en cuenta, operaciones de reserva y liberación de memoria dinámica.

EFICIENCIA de algoritmos- Tiempo de Ejecución

EFICIENCIA



La eficiencia de un programa se evalúa una vez que el programa es correcto. Está relacionado con los conceptos **MEMORIA** y **TIEMPO DE EJECUCIÓN**.

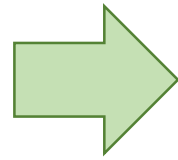
TIEMPO de EJECUCIÓN

Análisis empírico: requiere la implementación del programa, luego ejecutar el programa en la máquina y medir el tiempo consumido para su ejecución. La desventaja es que es dependiente de la máquina y hay que implementar la solución completa.

Análisis teórico: se analiza la estructura del programa, no es dependiente de la máquina. Se analiza el "peor caso"

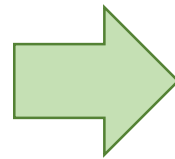
EFICIENCIA de algoritmos- Tiempo de Ejecución

¿Qué se tiene en cuenta?



Operaciones aritmético/lógicas : suma, resta, división, multiplicación, div, mod, asignación, <, >, =, <>, and, or not (no consideramos read, write, new, dispose).

1 UT (unidad de tiempo)



Estructuras de control: if, if else, for, while, repeat until

? UT (depende la estructura)

EFICIENCIA de algoritmos- Tiempo de Ejecución

If (condición) then
acciones

Tiempo de evaluar la condición + tiempo del cuerpo.
Si hay else, Tiempo de evaluar la condición +
 $\max(\text{then}, \text{else})$.

```
If (x < 5) then  
    x := x + 1
```

```
If (x < 5) then  
    x := x + 1  
Else  
    x := 0
```

For i:= 1 to cant do
acciones

$(3N + 2) +$
 $N(\text{cuerpo del for})$

```
For i:= 1 to 5 do  
    x := x + 1  
  
read (cant);  
for i:= 1 to cant do  
    x := x + 1
```

EFICIENCIA de algoritmos- Tiempo de Ejecución

While (condición) do
acciones

$C(N + 1) + N(\text{cuerpo del while})$. C representa el tiempo de evaluar la condición

```
i := 0;  
While (i < 5) do  
    x := x + 1;  
    i := i + 1
```

```
i := 0; ok := true;  
While (i < 5) and (ok = true) do  
    x := x + 1;  
    i := i + 1;
```

```
read(x)  
While (x < 5) do  
    x := x + 1;  
    i := i + 1
```

Repeat
acciones
Until (condición)

cuerpo del repeat + $C(N) + N(\text{cuerpo del repeat})$.

EFICIENCIA de algoritmos- Tiempo de Ejecución

Program ejemplo;

...

....

Begin

instrucción 1

instrucción 2

instrucción 3

instrucción 4

End.

Tiempo de Ejecución

Tiempo de instrucción 1 +

Tiempo de instrucción 2 +

Tiempo de instrucción 3 +

Tiempo de instrucción 4 +

EFICIENCIA de algoritmos- Tiempo de Ejecución

Tiempo de Ejecución

```
Program uno;  
var  
  i, aux: integer;  
  
Begin  
  read(aux);  
  for i:= 3 to 22 do  
    begin  
      aux:= 2;  
      aux:= aux + 1;  
    end  
  end,  
end;
```

$$T(1) = T(\text{read}) \quad 0 \text{ UT}$$

$$T(2) = T(\text{for})$$

$$3N + 2 + N \text{ (cuerpo)}$$

$$N = 20 \quad (ls-li + 1)$$

$$3(20) + 2 + 20 \text{ (cuerpo)}$$

$$\text{cuerpo} = 1 + 2 = 3$$

$$3(20) + 2 + 20 (3) \quad 122 \text{ UT}$$

$$TE = 0 + 122 = 122 \text{ UT}$$

EFICIENCIA de algoritmos- Tiempo de Ejecución

Tiempo de Ejecución

```
Program dos;  
var  
  i,aux: integer;  
  
Begin  
  read(i); aux:= 0;  
  while (i <> 9) do  
    begin  
      aux:= aux + i;  
      read(i);  
    end  
  end,  
end;
```

$$T(1) = T(\text{read}) \quad T(2) = T(\text{asig})$$

0 UT **1 UT**

$$T(3) = T(\text{while})$$

$C(N + 1) + N$ (cuerpo)

$C = 1$ $N = ?$

cuerpo = 2

$$1(N + 1) + N (2)$$

$$TE = 0 + 1 + 3N + 1 \text{ UT}$$

$$TE = 2 + 3N \text{ UT}$$

$$N + 1 + 2N \text{ UT}$$

$$3N + 1 \text{ UT}$$

Program tres;

var

i,aux, aux2: integer;

Begin

aux:= 100;

for i:= 3 to 8 do

begin

read (aux2);

if (aux2 >= 5) then

begin

j:= 1;

while (j < 10) do begin

aux2:= aux2 + i + aux2 * 2;

j:= j + 3; end;

end

else

for i:= 1 to 2 do aux2:= 1;

end;

End;

TE= 1 + 158 UT

Tiempo de Ejecución

$T(1) = T(\text{asig})$ **1 UT**

$T(2) = T(\text{for}) = 3N + 2 + N (\text{cuerpo})$

$N = 6$ cuerpo =

$T(3) = T(\text{read})$ **0 UT**

$T(4) = T(\text{ifelse})$

$T(4) = C + \text{Max}(\text{if}, \text{else})$

$T(4) = 1 + \text{Max}(\text{if}, \text{else})$

$T(4) = \text{Max} (1+\text{while}; \text{for})$

$\text{While } M+1 + M(\text{cuerpo}) = 4 + 3(6)$

22

$\text{for } 3L+2 + L(\text{cuerpo}) = 8 + 2(1)$

10

$T(2)=3*6+2+6(23) = 158$ 1 + 22 UT