Algoritmos y su Complejidad

Marco Antonio Heredia

Posgrado en
Optimización

Universidad
Autónoma
Metropolitana
Casa abierta al tiempo Azcapotzalco

¿Qué es un algoritmo?

Informalmente:

Conjunto finito de pasos que, dada la entrada adecuada, obtiene siempre la solución de un problema particular.

Ejemplos de Algoritmos

Cualquier receta de cocina.

Entrada: Huevos, sartén, aceite, pala, estufa, sal.

1.- Poner estufa a fuego medio.
2.- Pon la sartén sobre el fuego con un poco de aceite.
3.- Rompe los huevos y vaciarlos sobre la sartén.
4.- Sal al gusto y revolver por 4 minutos con la pala.

Salida: Huevos revueltos.

Ejemplos de Algoritmos

Proceso de vestirse.

Entrada: Persona desnuda y ropa.

1.- Poner calcetín izquierdo.
2.- Poner calcetín derecho.
3.- Poner calzones.
... (pasos adicionales) ...

Salida: Persona vestida.

Ejemplos de Algoritmos

$$\begin{array}{r} 36 \\ +\ 47 \\ \hline 83 \end{array}$$

Suma de 2 enteros.

Entrada: Dos enteros (en base 10).

1. Sumar unidades, y si > 10, escribir sólo unidades y guardar 1 acarreo para siguiente paso.
2.- Sumar decenas y acarreo (si lo hubo), escribir sólo unidades y si > 10, guardar 1 acarreo para siguiente paso.
(continuar recorriendo posiciones)...

Salida: La suma de ambos números (base 10).

No es algoritmo cuando...

- Si el conjunto de pasos no es finito, entonces no es Algoritmo.

- Si el conjunto de pasos no garantiza llegar a una solución, entonces no es Algoritmo. (Algunos de estos métodos reciben el nombre de Heurísticas y tienen su propia importancia)

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

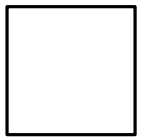
-1	7	5	2	15	8	-1	11	31	-6	18	9	3	17
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

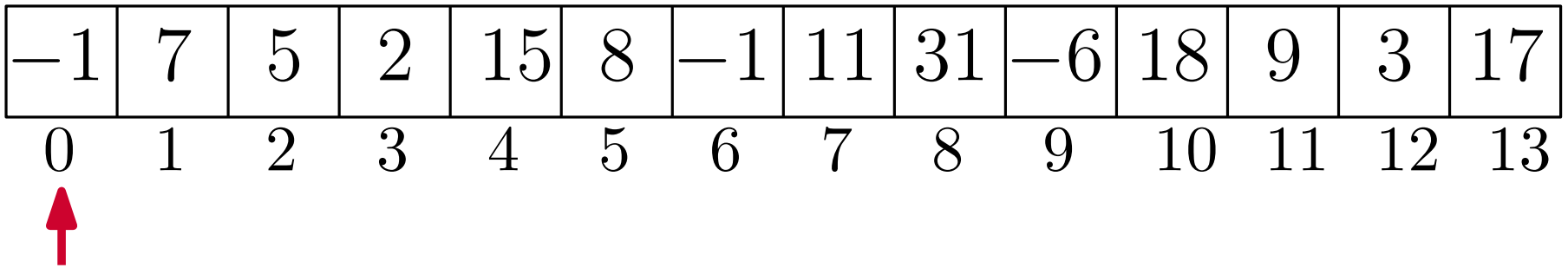
-1	7	5	2	15	8	-1	11	31	-6	18	9	3	17
0	1	2	3	4	5	6	7	8	9	10	11	12	13

max



Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

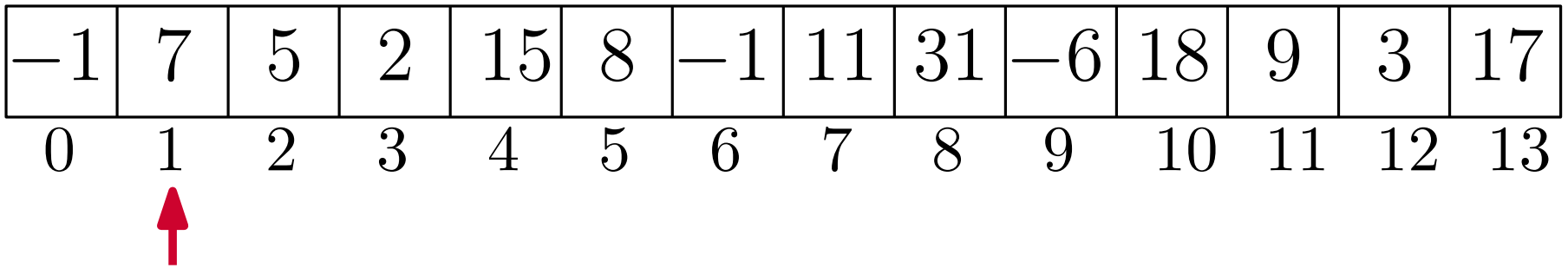


max

−1

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

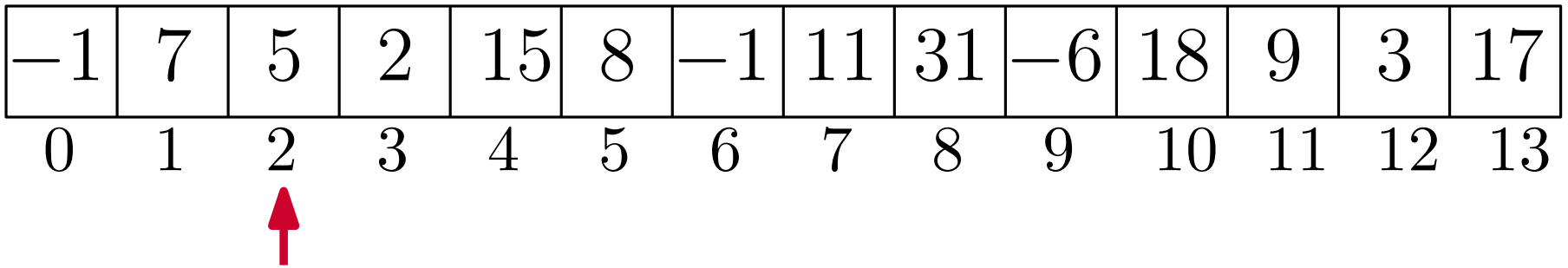


max

7

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

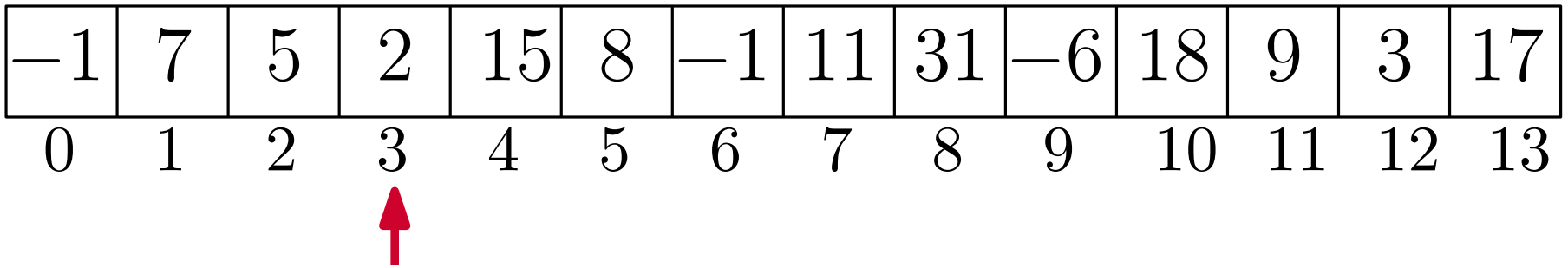


max

7

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

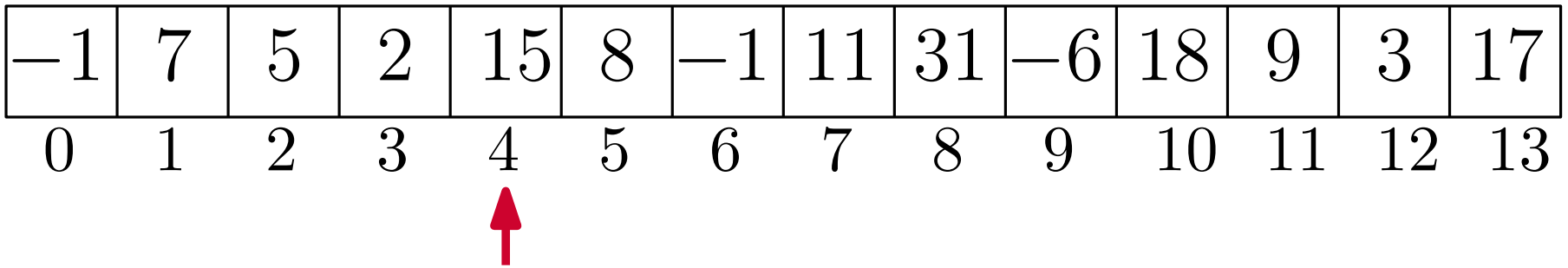


max

7

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

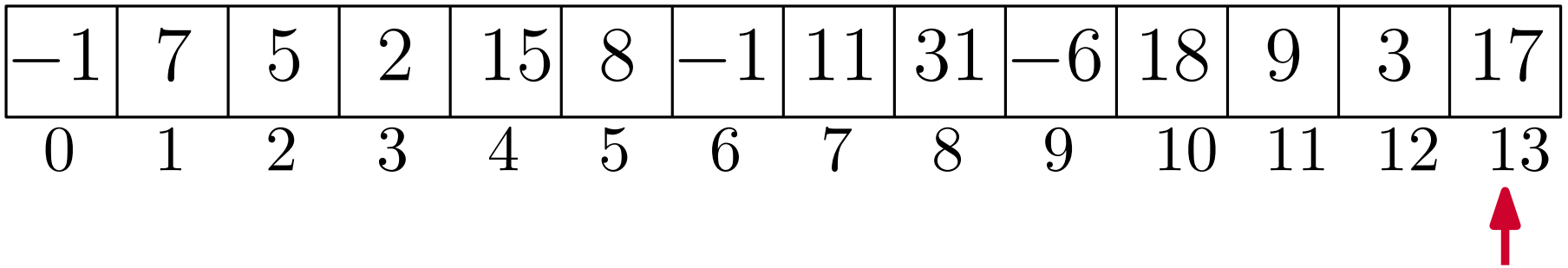


max

15

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.



max

31

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

```
int max = A[0];

for (i=1; i < n;  i++){
    if (A[i] > max){
      max = A[i];
    }
}

printf("El mayor es %d", max);
```

Máximo o mínimo

Problema: Dado un arreglo A de n números enteros encuentre el elemento máximo.

-1	7	5	2	15	8	-1	11	31	-6	18	9	3	17
0	1	2	3	4	5	6	7	8	9	10	11	12	13

¿Es la única forma?

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos medir el tiempo que se tarda en ejecutar cada uno en una computadora.

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos medir el tiempo que se tarda en ejecutar cada uno en una computadora.

  - ¿Qué pasa si se ejecuta otro programa (e.g. antivirus) cuando se ejecutó el algoritmo A2?

  - ¿Qué pasa si uno de los dos algoritmos está pobremente implementado?

  - ¿Qué pasa si un algoritmo fue ejecutado en su "mejor" caso y ese corresponde al "peor" caso del otro algoritmo?

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos medir la cantidad de recursos utilizados por cada uno.

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos medir la cantidad de recursos utilizados por cada uno.

  - ¿Qué pasa si el algoritmo que ocupa menos recursos tarda 100 veces más en terminar?

  - ¿Qué pasa si el algoritmo más eficiente en recursos es muy difícil de implementar y sólo ahorra el 2% de recursos que el otro algoritmo?

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos contar el número de instrucciones de cada uno.

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos contar el número de instrucciones de cada uno.

A1

```
int a = 12;
int res =0;
res = res + a;
res = res + a;
res = res + a;
res = res + a;
res = res + a;
```

A2

```
int res;
res = 12 * 5;
```

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos contar el número de instrucciones de cada uno.

A1

```
int res;
res = (10 + 2) * 5;
```

A2

```
int res;
res = 12.0 * 5.0;
printf("El");
printf("resultado");
printf("escrito.");
```

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Podemos contar el número de instrucciones de cada uno.

A1

```
int res;
res = (10 + 2) * 5;
```

A2

```
int res;
res = 12.0 * 5.0;
printf("El");
printf("resultado");
printf("escrito.");
```

A2 tiene más instrucciones, pero muchas son *superfluas*. El cálculo de A1 son dos operaciones de enteros, en A2 sólo es una operación, pero es de punto flotante.

Comparación entre algoritmos

Dados 2 algoritmos, A1 y A2, que resuelven el mismo problema. ¿Cuál de ellos es el mejor?

- Dado que ninguna de las 3 métricas sugeridas es perfecta, nos quedaremos con la menos mala.
- Vamos a contar el número de operaciones / instrucciones, pero tomando en cuenta algunas consideraciones para solventar sus inconvenientes.

Consideraciones para contar operaciones

- Llamaremos *instrucción simple* a aquella que se puede ejecutar de una sola vez, sin llamadas a otros métodos. Por ejemplo, la declaración de una variable, la asignación de un valor o la evaluación de una expresión aritmética.

```
int a = ((38 * 64) / 25) +3;

float b;

b = (5 - (3.1416 * a)) / 17;
```

1 instr. simple = 1 operación

Consideraciones para contar operaciones

- Contaremos siempre el número de operaciones que realiza el algoritmo en su *peor caso*.

Consideraciones para contar operaciones

- Contaremos siempre el número de operaciones que realiza el algoritmo en su *peor caso*.

  - Saber cuántas operaciones hace un algoritmo en su mejor caso da poca información sobre su comportamiento en general.

  - Considerar el peor caso, al menos da una cota superior del numéro de operaciones que realizará el algoritmo.
  Tu algoritmo a lo más hará tantas operaciones.

operaciones de una secuencia

instrucción simple 1;
instrucción simple 2;
instrucción simple 3;
...
instrucción simple k;

operaciones de una secuencia

instrucción simple 1;
instrucción simple 2;
instrucción simple 3;
...
instrucción simple k;

k operaciones

operaciones de una secuencia

instrucción simple 1;
instrucción simple 2;
instrucción simple 3;
...
instrucción simple k;

k operaciones

```
int a = 5;
int b = 8;
int c;
c =((45 * (17+b)) + a) / 2;
```

4 operaciones

operaciones de un ciclo

```
ciclo {
    |
    |
    |
}
```

$\longleftarrow$ p número de vueltas

m operaciones dentro

operaciones de un ciclo

```
ciclo {
    |
    |
    |
}
```

$\longleftarrow$ p número de vueltas

m operaciones dentro

$p * m$ operaciones

operaciones de un ciclo

```
ciclo {
    |
    |
    |
}
```

p número de vueltas

m operaciones dentro

$p * m$ operaciones

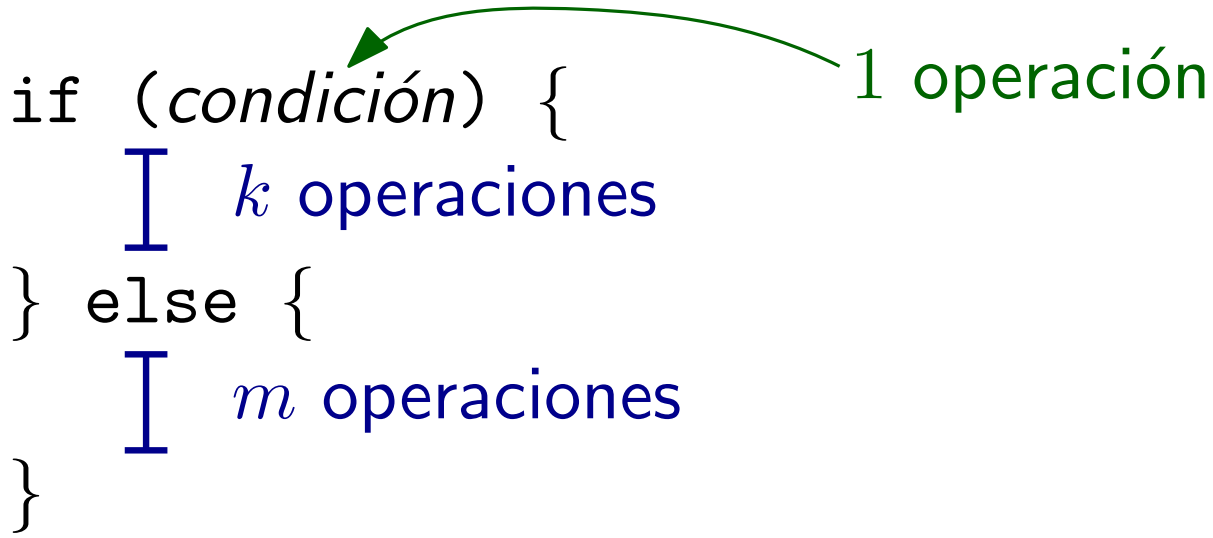
```
for (i =1; i <= 23; i++){
  c= 34;
  d= 14;
  m = 67 + i;
  x = c + d + m;
}
```

23 vueltas

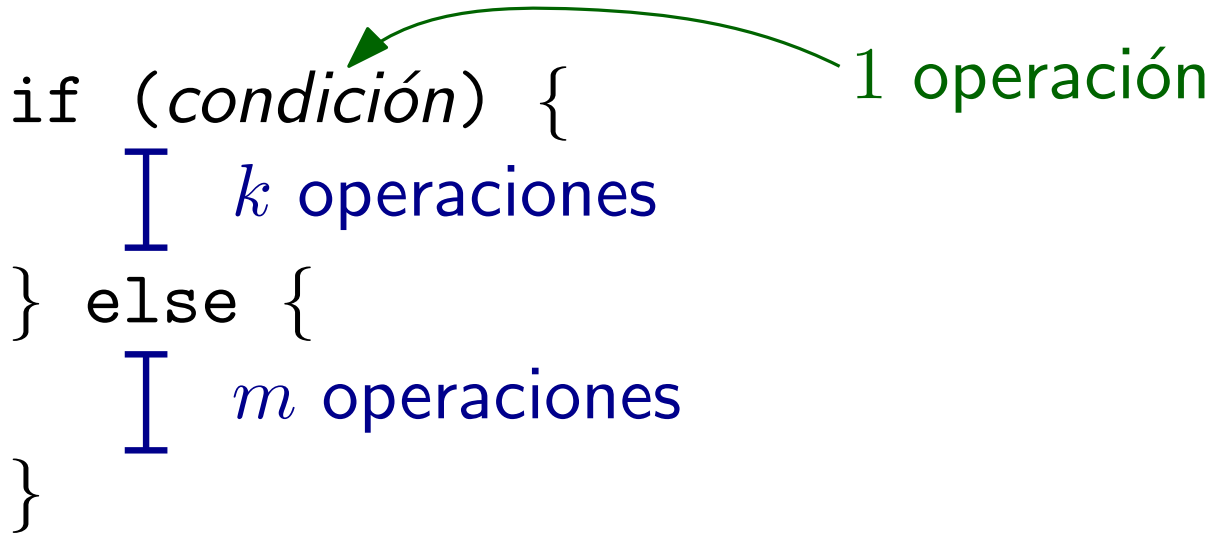
4 operaciones

$23 * 4 = 92$ operaciones

operaciones de un condicional



operaciones de un condicional



Como tomamos el peor caso:

$\max(k,m) + 1$

operaciones de un condicional

```
if (condición) {
    ⟦ k operaciones
} else {
    ⟦ m operaciones
}
```

1 operación

Como tomamos el peor caso:

$\max(k,m) + 1$

```
if ( a > 20) {
   c= 48;
   x= c +2;
} else {
   c= 34;
   y= 18;
   z= (c*y)-5;
}
```

2 operaciones

3 operaciones

$\max(2,3) + 1 =$ 4 operaciones