



Básicos

N Sesión by Abii Snn

Dado un vector **ordenado** y un número,
imprime 1 si el número existe en el arreglo.

15	15	20	22	24	25	26	32	40
----	----	----	----	----	----	----	----	----

Número = 26 ----> Yes

15	15	20	22	24	25	26	32	40
----	----	----	----	----	----	----	----	----

Número = 100 ----> No

Posible solución

→ Buscar elemento por elemento y preguntar si ese elemento es igual al que busco. $O(n)$



EODDING TIME

Dado un vector **ordenado** y una serie de números, dime si los números existen en el arreglo (query).

15	15	20	22	24	25	26	32	40
----	----	----	----	----	----	----	----	----

Número = 100----> No

Número = 22 ---> Yes

Número = 50----> No

Número = 26 ----> Yes

¿Cuál sería la complejidad?

Posible solución

→ Para cada número, buscar elemento por elemento y preguntar si ese elemento es igual al que busco.

$$O(n * q)$$

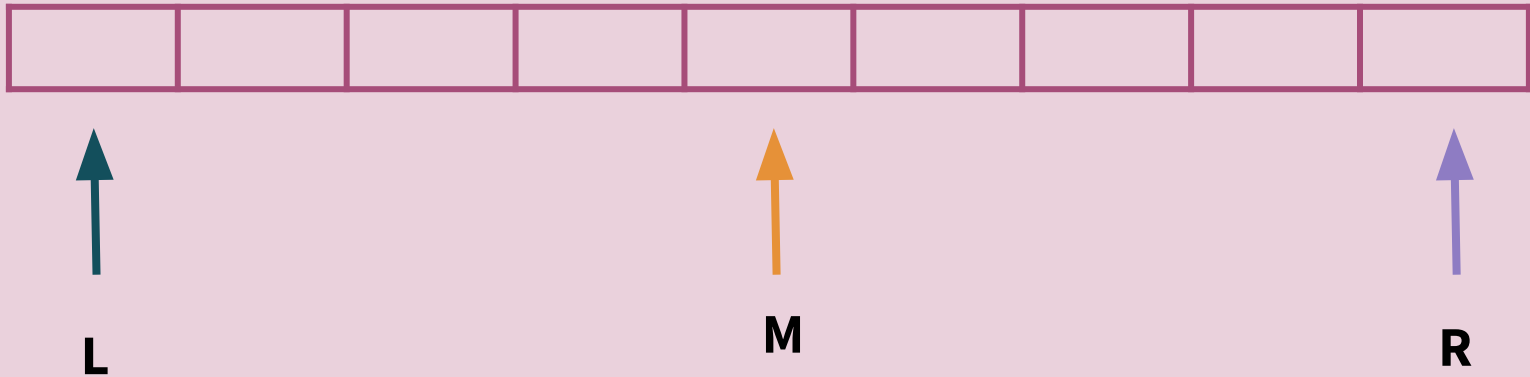
n: Tamaño del arreglo

q: Número de preguntas



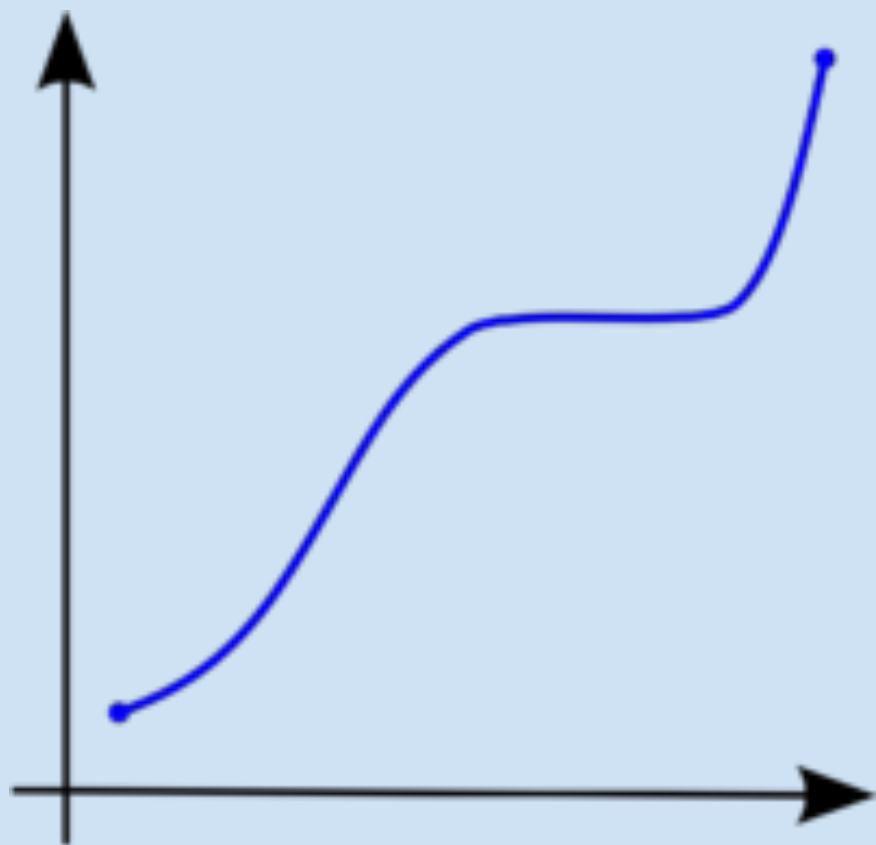
EODING TIME

Búsqueda binaria



¿Qué sabemos? / **Requisitos**

**secuencia monótona
creciente.**



Es por eso que quizá, has escuchado que para hacer una **búsqueda binaria** el vector debe estar ordenado.

Para el problema...

¿De qué me sirve que el vector
esté ordenado?

Buscando a 26

15	15	20	22	24	25	26	32	40
F	F	F	F	F	F	V		

La **búsqueda binaria** te dice:
si ya tienes una secuencia
monótona creciente, no busques
en todos los elementos, busca en
segmentos que te sirvan.

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

¿Qué conocemos?
Left y Right, podemos sacar middle

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

¿De qué nos sirve conocer a **middle**?

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

¿De qué nos sirve conocer a **middle**?
Nos ayuda a decidir a qué **dirección** movernos.

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] < 26$?

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] < 26$?

Como es un arreglo ordenado, el número que
estoy buscando debería estar a la derecha.

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] < 26$?

BUSCO A LA DERECHA. (L cambia)

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
`vector[middle] > 26` ?

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] > 26$?

Como es un arreglo ordenado, el número
debería estar a la izquierda.

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] > 26$?

BUSCO A LA IZQUIERDA. (Cambia R)

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
`vector[middle] == 26` ?

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

Si estoy buscando a 26, ¿qué pasa si
 $\text{vector}[\text{middle}] == 26$?

ENCONTRÉ LA RESPUESTA.



CASOS:

$\text{vector}[m] == \text{target}$:

SALIR

$\text{vector}[m] < \text{target}$:

Moverme a la derecha.

$\text{vector}[m] > \text{target}$:

Moverme a la izquierda.

Ejemplo:

Cuando el número **si** existe en el arreglo.

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

$l = 0, r = 8$

$m = (l + r) / 2 = (0 + 8) / 2 = 4$

$24 < 26$: **ir a la derecha**

$l = \text{middle} + 1 = 5$

Buscando a 26

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

$$l = 5, r = 8$$

$$m = (l + r) / 2 = (5 + 8) / 2 = 6$$

26 == 26 : **SALIR**

Cuando el número no existe
en el arreglo

Buscando a 19

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

$l = 0, r = 8$

$m = (l + r) / 2 = (0 + 8) / 2 = 4$

$24 > 19$: **ir a la izquierda**

$r = \text{middle} - 1 = 3$

Buscando a 19

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Left



Middle



Right

$l = 0, r = 3$

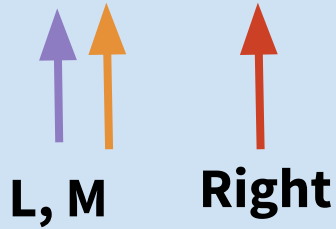
$m = (l + r) / 2 = (0 + 3) / 2 = 1$

$15 < 19$: **ir a la derecha**

$l = \text{middle} + 1 = 2$

Buscando a 19

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



$$l = 2, r = 3$$

$$m = (l + r) / 2 = (2 + 3) / 2 = 2$$

$20 > 19$: **ir a la izquierda**

$$r = \text{middle} - 1 = 1$$

Buscando a 19

15	15	20	22	24	25	26	32	40
0	1	2	3	4	5	6	7	8



Right



Left

$l = 2, r = 1$

$r < l$ No puede pasar

¿Cuál es la complejidad?

$O(\log(N))$

N: tamaño del vector

Posible solución: Búsqueda binaria

→ Para cada número, usar búsqueda binaria para saber si el vector tiene el número.

$$O(q * \log(n))$$

q: Número de queries

n: Tamaño del arreglo



EODING TIME

¿Qué puede pasar con esta
fórmula?

$$\text{mid} = (l + r) / 2$$



**SE PUEDE
DESBORDAR LA
VARIABLE.**

Sí mid es un entero

$$\text{mid} = (l + r) / 2$$

$$\text{mid} = (1\text{e}18 + 1\text{e}18)/2$$

Solución:

$$\text{mid} = L + (R - L) / 2$$

Motivación

- Entrevistas
- Tema súper básico
- Gran ayuda para muchos problemas

Tip personal:

Siempre piensa en los 3 casos que analizamos, pregúntate:
¿A dónde debo caminar/buscar?

PROBLEMA

Dado un vector **ordenado** y un número, dime la frecuencia del número en el arreglo.

15	15	20	22	24	25	26	32	40	40	40	40	50	55
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Número = 15, frecuencia = 2

Número = 40, frecuencia = 4

Número = 20, frecuencia = 1

Número = 99, frecuencia = 0

Possible solution

→ Pass through all the elements and count how many are equal to the one I am looking for.

$O(n)$

n : Size of the array

LOWER BOUND

Primer elemento que **no** se compara como **menor** al número que buscamos.

LOWER BOUND

Si buscamos el lower bound de 40:
Es como si empezáramos desde el principio
preguntando: ¿Eres menor a 40?
Lower bound retorna el **PRIMER FALSO**.

15	15	20	22	24	25	26	32	40	40	40	40	50	55
V	V	V	V	V	V	V	V	F	F	F	F	F	F

LOWER BOUND

Si buscamos el lower bound de 45:
Es como si empezáramos desde el principio preguntando: ¿Eres menor a 45?
Lower bound retorna el **PRIMER FALSO**.

[illegible]

LOWER BOUND

Primer elemento que no se compara como **menor** al número que buscamos.

15	15	20	22	24	25	26	32	40	40	40	40	50	55
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Lower bound de 39: 40 (primer 40)

Lower bound de 40: 40 (primer 40)

Lower bound de 45: 50


Lower bound de 60: No existe

PENSEMOS EN LOS 3 CASOS

¿Qué hago cuando...
 $v[\text{mid}] > \text{target}$

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ejemplo, si busco lower
de 40 y el middle es 50, $\Rightarrow 50 > 40$



Middle

¿Qué hago cuando...

$v[mid] > target$: camino a la izquierda

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ejemplo, si busco 40 y $50 > 40$



Middle

¿Qué hago cuando...
 $v[\text{mid}] < \text{target}$

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Middle

$24 < 40$

¿Qué hago cuando...

$v[\text{mid}] < \text{target}$: camino a la derecha

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Middle

$24 < 40$

¿Qué hago cuando...
`v[mid] == target`:

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13

¿Qué hago cuando...
 $v[\text{mid}] == \text{target}$:

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Middle

¿Qué hago cuando...

$v[mid] == target$: camino a la izquierda

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Middle

Ejemplo:

Cuando el lower bound del número si existe.

Buscando lower bound de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Left



Middle



Right

$l = 0, r = 13$

$m = l + (r - l) / 2 = 0 + (13 - 0) / 2 = 6$

$26 < 40$: **ir a la derecha**

$l = \text{middle} + 1 = 7$

Buscando lower bound de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Left



Middle



Right

$$l = 7, r = 13$$

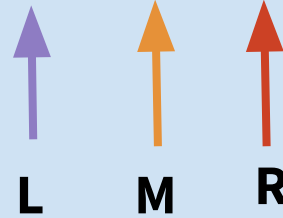
$$m = l + (r - l) / 2 = 7 + (13 - 7) / 2 = 10$$

40 == 40 : **ir a la izquierda**

$$r = \text{middle} - 1 = 9$$

Buscando lower bound de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



$$l = 7, r = 9$$

$$m = l + (r - l) / 2 = 7 + (9 - 7) / 2 = 8$$

40 == 40 : **ir a la izquierda**

$$r = \text{middle} - 1 = 7$$

Buscando lower bound de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



L M R

$$l = 7, r = 7$$

$$m = l + (r - l) / 2 = 7 + (7 - 7) / 2 = 7$$

$32 < 40$: **ir a la derecha**

$$l = \text{middle} + 1 = 8$$

Buscando lower bound de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



R



L

$l = 8, r = 7$

$l > r$, esto no puede pasar

l tiene la respuesta

Ejemplo:

Cuando el lower bound del número no existe.

Buscando lower bound de 60

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Left



Middle



Right

$l = 0, r = 13$

$m = l + (r - l) / 2 = 0 + (13 - 0) / 2 = 6$

$26 < 60$: **ir a la derecha**

$l = \text{middle} + 1 = 7$

Buscando lower bound de 60

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Left



Middle



Right

$l = 7, r = 13$

$m = l + (r - l) / 2 = 7 + (13 - 7) / 2 = 10$

$40 < 60$: **ir a la derecha**

$l = \text{middle} + 1 = 11$

Buscando lower bound de 60

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13

A diagram illustrating a binary search step. Three arrows point upwards to the bottom row of the table: a purple arrow labeled 'L' points to index 11, an orange arrow labeled 'M' points to index 12, and a red arrow labeled 'R' points to index 13.

$l = 11, r = 13$

$m = l + (r - l) / 2 = 11 + (13 - 11) / 2 = 12$

$50 < 60$: **ir a la derecha**

$l = \text{middle} + 1 = 13$

Buscando lower bound de 60

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



$l = 13, r = 13$

$m = l + (r - l) / 2 = 13 + (13 - 13) / 2 = 13$

$55 < 60$: **ir a la derecha**

$l = \text{middle} + 1 = 14$

Buscando lower bound de 60

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



$l = 14, r = 13$

$l > r$, esto no puede pasar
retornamos -1.



COOLING TIME

UPPER BOUND

Primer número **mayor** a NUMBER.

UPPER BOUND

Si buscamos el upper bound de 40:

Es como si empezáramos desde el principio preguntando: ¿Eres mayor a 40?

Upper bound retorna el **PRIMER VERDADERO**.

[illegible]

UPPER BOUND

Si buscamos el upper bound de 23:

Es como si empezáramos desde el principio preguntando: ¿Eres mayor a 23?

Upper bound retorna el **PRIMER VERDADERO.**

[illegible]

UPPER BOUND

Si buscamos el upper bound de 100:

Es como si empezáramos desde el principio preguntando: ¿Eres mayor a 100?

Upper bound retorna el **PRIMER VERDADERO**.

15	15	20	22	24	25	26	32	40	40	40	40	50	55
F	F	F	F	F	F	F	F	F	F	F	F	F	F

En este caso la respuesta es -1, porque no existe.

UPPER BOUND

Primer número **mayor** a NUMBER.

15	15	20	22	24	25	26	32	40	40	40	40	50	55
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Upper bound de 39: 40

Upper bound de 40: 50

Upper bound de 45: 50

Upper bound de 60: No existe



COOLING TIME



**PROGRAMA UPPER
BOUND.
PIENSA EN LOS 3
CASOS.**

Posible solución al problema de frecuencia: Búsqueda binaria

→ Usar dos binarias:

- Lower bound. // $O(\log(n))$
 - Upper bound. // $O(\log(n))$
- $O(\log(n) + \log(n))$

n: Tamaño del arreglo

Buscando frecuencia de 40

15	15	20	22	24	25	26	32	40	40	40	40	50	55
0	1	2	3	4	5	6	7	8	9	10	11	12	13



Lower bound



Upper bound

$$\text{Frecuencia} = \text{Upper bound} - \text{lower bound} = 12 - 8 = 4$$



EODING TIME

```
27  int main() {  
28      // Sacar la frecuencia de un número  
29      int n; cin >> n;  
30      vector<int> v(n);  
31      for(int i = 0; i < n; i++) cin >> v[i];  
32  
33      int number; cin >> number;  
34      int low = lowerBound(v, number);  
35      int upper = upperBound(v, number);  
36      cout << upper - low << endl;  
37  }
```

Dado un número **n**, imprime verdadero si tiene raíz cuadrada exacta.

No puedes usar funciones de sqrt.

$$0 < n < 10^9$$

Número = 16, si, porque su raíz es 4

Número = 20, no, su raíz es 4.47

Posible solución

→ Iterar desde 1 hasta n preguntando si:
 $i * i == n$

Complejidad $O(n)$
TLE

Posible solución: Búsqueda binaria

→ Usar búsqueda binaria para encontrar la respuesta.

Complejidad $O(\log(n))$



CASOS:

$m * m == \text{target}$:

SALIR

$m * m < \text{target}$:

Moverme a la derecha.

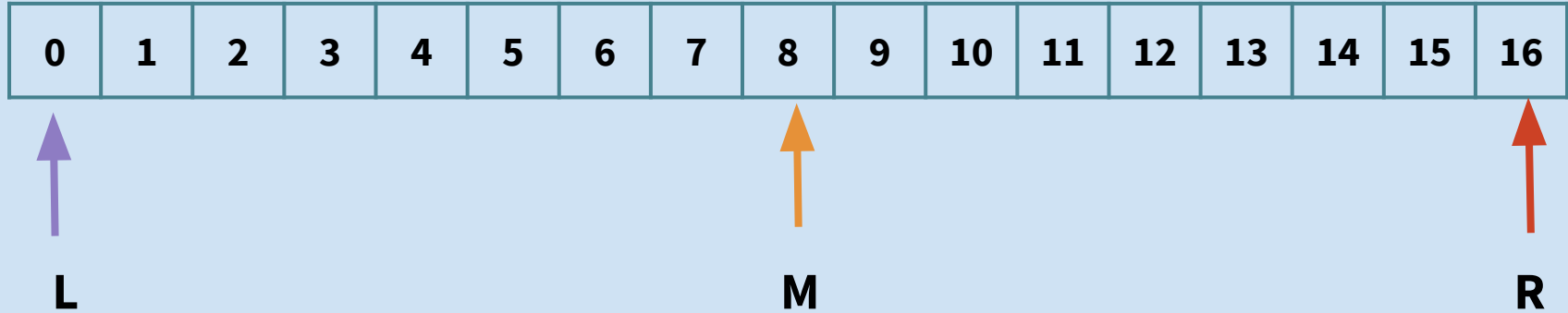
$m * m > \text{target}$:

Moverme a la izquierda.

Ejemplo:

Cuando el número tiene raíz entera.

Buscando respuesta para 16



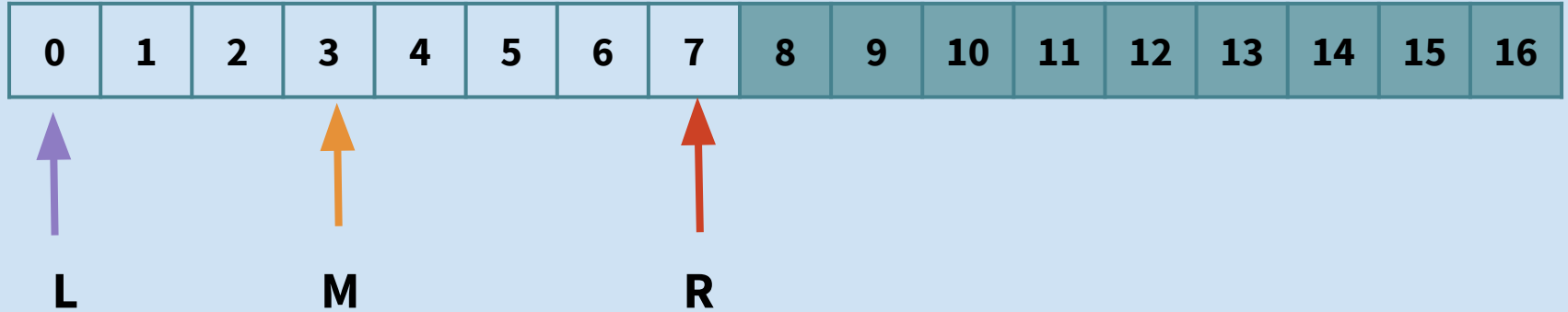
$$l = 0, r = 16$$

$$m = l + (r - l) / 2 = 0 + (16 - 0) / 2 = 8$$

$8 * 8 > 16$: **ir a la izquierda**

$$r = \text{middle} - 1 = 7$$

Buscando respuesta para 16



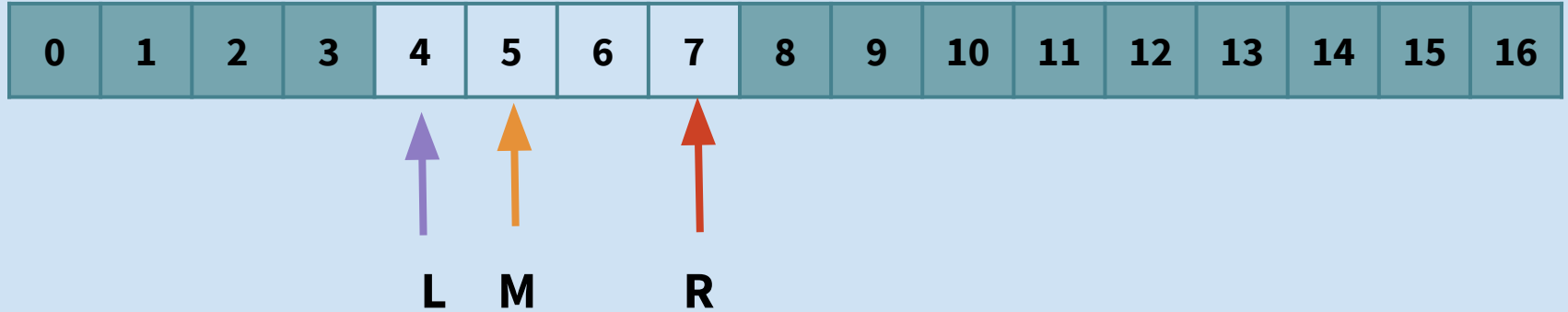
$$l = 0, r = 7$$

$$m = l + (r - l) / 2 = 0 + (7 - 0) / 2 = 3$$

$3 * 3 < 16$: **ir a la derecha**

$$l = \text{middle} + 1 = 4$$

Buscando respuesta para 16



$$l = 4, r = 7$$

$$m = l + (r - l) / 2 = 4 + (7 - 4) / 2 = 5$$

$5 * 5 > 16$: **ir a la izquierda**

$$r = \text{middle} - 1 = 4$$

Buscando respuesta para 16

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----



LMR

$$l = 4, r = 4$$

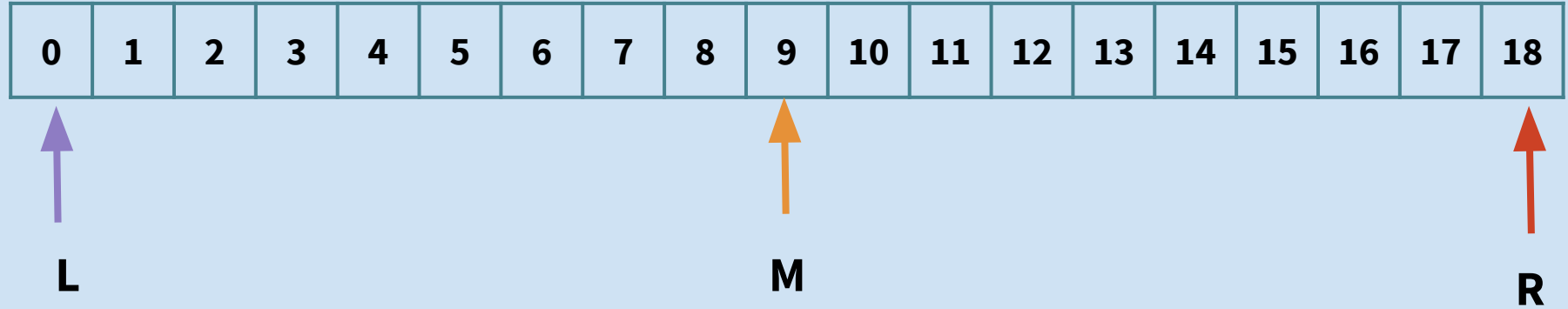
$$m = l + (r - l) / 2 = 4 + (4 - 4) / 2 = 4$$

$4 * 4 == 16$: Es la respuesta

Ejemplo:

Cuando el número no tiene
raíz entera.

Buscando respuesta para 18



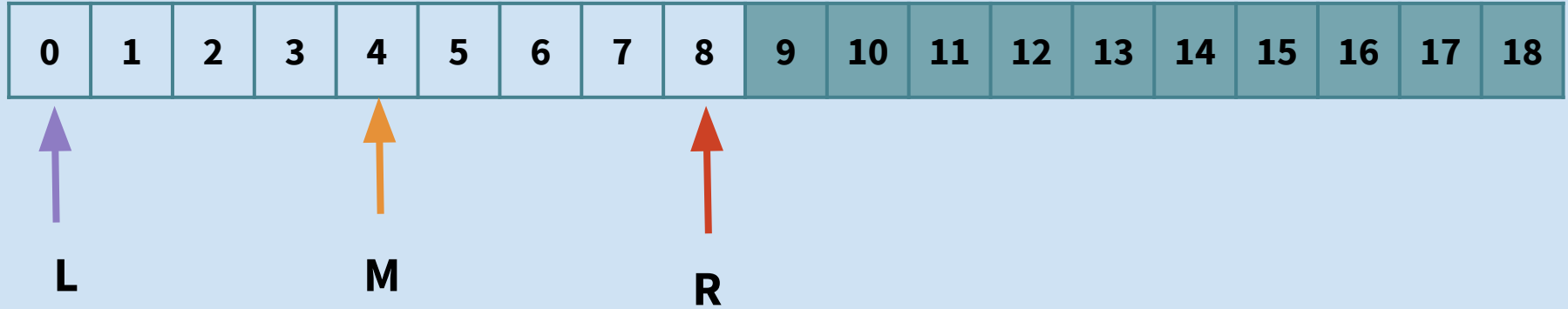
$$l = 0, r = 18$$

$$m = l + (r - l) / 2 = 0 + (18 - 0) / 2 = 9$$

$9 * 9 > 18$: **ir a la izquierda**

$$r = \text{middle} - 1 = 8$$

Buscando respuesta para 18



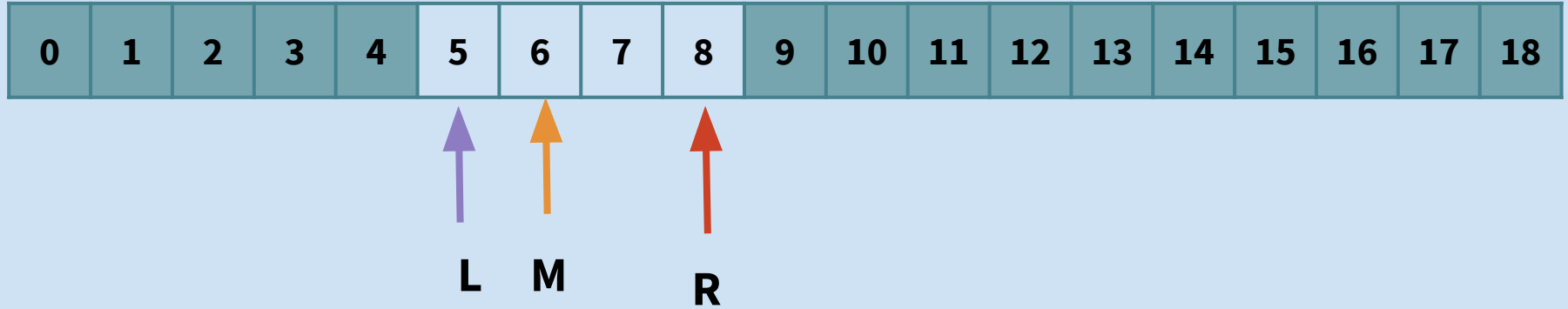
$$l = 0, r = 8$$

$$m = l + (r - l) / 2 = 0 + (8 - 0) / 2 = 4$$

$4 * 4 < 18$: **ir a la derecha**

$$l = \text{middle} + 1 = 5$$

Buscando respuesta para 18



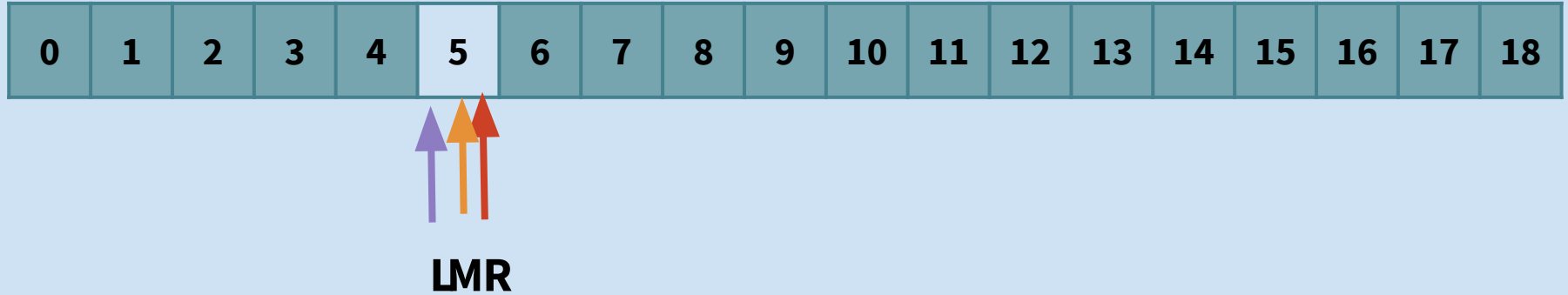
$$l = 5, r = 8$$

$$m = l + (r - l) / 2 = 5 + (8 - 5) / 2 = 6$$

$6 * 6 > 18$: **ir a la izquierda**

$$r = \text{middle} - 1 = 5$$

Buscando respuesta para 18



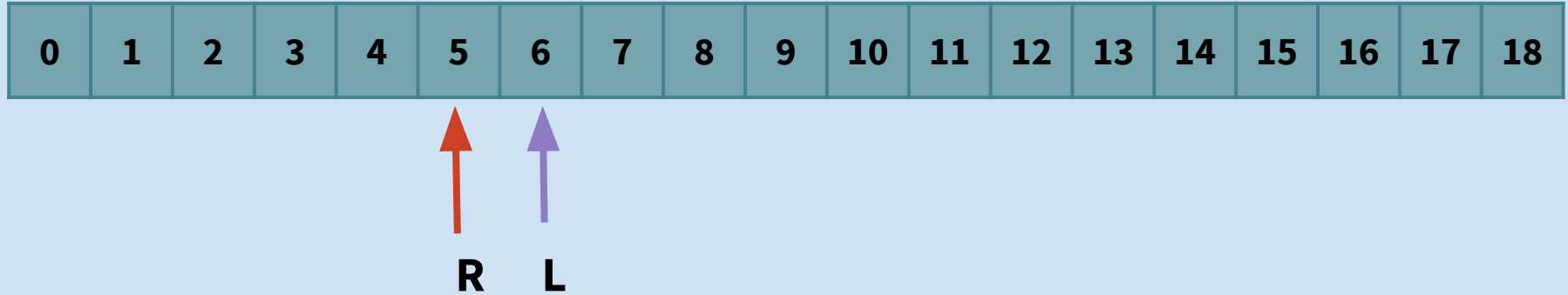
$$l = 5, r = 5$$

$$m = l + (r - l) / 2 = 5 + (5 - 5) / 2 = 5$$

$5 * 5 > 18$: **ir a la derecha**

$$l = \text{middle} + 1 = 6$$

Buscando respuesta para 18



$l = 6, r = 5$

$l > r$, esto no puede pasar, aquí termina.

Retornamos -1.

Observaciones para implementación:

No es necesario crear el arreglo hasta n , trabajamos con middle.



EODING TIME