Búsqueda binaria

Miguel Ortiz

Programación competitiva para ICPC

Octubre 2023

Problema de ejemplo

Dado un arreglo **ordenado** de enteros, indicar si contiene un elemento *val* o no.

Problema de ejemplo

Dado un arreglo **ordenado** de enteros, indicar si contiene un elemento *val* o no.

Algoritmo:

- Revisamos el elemento del medio
    - Si es *val*, terminamos
    - Si es mayor que *val*, buscamos en la mitad izquierda
    - Si es menor que *val*, buscamos en la mitad derecha

Problema de ejemplo

Dado un arreglo **ordenado** de enteros, indicar si contiene un elemento *val* o no.

Algoritmo:

- Revisamos el elemento del medio
  - Si es *val*, terminamos
  - Si es mayor que *val*, buscamos en la mitad izquierda
  - Si es menor que *val*, buscamos en la mitad derecha
- Hacemos lo mismo en las mitades más pequeñas hasta encontrar el número o hasta que no queden más elementos

Problema de ejemplo

Dado un arreglo **ordenado** de enteros, indicar si contiene un elemento val o no.

Algoritmo:

- Revisamos el elemento del medio
  - Si es val, terminamos
  - Si es mayor que val, buscamos en la mitad izquierda
  - Si es menor que val, buscamos en la mitad derecha
- Hacemos lo mismo en las mitades más pequeñas hasta encontrar el número o hasta que no queden más elementos
- Primero hay n candidatos, luego $\frac{n}{2}$, luego $\frac{n}{4}$, etc.
- $O(\log n)$

Problema de ejemplo

$val = 22$

1	4	5	8	11	19	21	22	25	29	31	37	42
lo						m						hi

Problema de ejemplo

$val = 22$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo		m	m		hi

Problema de ejemplo

$val = 22$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo		m			hi

Problema de ejemplo

$val = 22$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo	hi	×	×	×	×
							m					

Problema de ejemplo

$val = 22$

1	4	5	8	11	19	21	**22**	25	29	31	37	42
×	×	×	×	×	×	×	lo	hi	×	×	×	×
							m					

Problema de ejemplo

$val = 23$

1	4	5	8	11	19	21	22	25	29	31	37	42
lo						m						hi

Problema de ejemplo

$val = 23$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo		m			hi

Problema de ejemplo

$val = 23$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo	hi	×	×	×	×
							m					

Problema de ejemplo

$val = 23$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	×	lo	×	×	×	×
								hi				
								m				

Problema de ejemplo

$val = 23$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	×	×	×	×	×	×
							hi	lo				

Problema de ejemplo

```
... // leer n, val, arreglo a
int lo = 0, hi = n-1;
int res = -1;
while (lo <= hi) {
  int mid = (lo + hi) / 2;
  if (a[mid] == val) {
    res = mid;
    break;
  }
  else if (val < a[mid]) {
    hi = mid - 1;
  }
  else { // val > a[mid]
    lo = mid + 1;
  }
}
// val esta en la posicion res, -1 si no esta
```

Forma general

- Trabajar sobre un rango de enteros
- $f(x) = \begin{cases} 1 & \text{si } x \text{ cumple la propiedad} \\ 0 & \text{si } x \textbf{ no} \text{ cumple la propiedad} \end{cases}$
- $f(x)$ es monótona $\star$

Forma general

- Trabajar sobre un rango de enteros
- $f(x) = \begin{cases} 1 & \text{si } x \text{ cumple la propiedad} \\ 0 & \text{si } x \textbf{ no} \text{ cumple la propiedad} \end{cases}$
- $f(x)$ es monótona $\star$

$$val = 22 \quad f(x) = \begin{cases} 1 & \text{si } x \geq val \\ 0 & \text{si } x < val \end{cases}$$

1	4	5	8	11	19	21	22	25	29	31	37	42
0	0	0	0	0	0	0	1	1	1	1	1	1

Forma general

- Trabajar sobre un rango de enteros
- $f(x) = \begin{cases} 1 & \text{si } x \text{ cumple la propiedad} \\ 0 & \text{si } x \textbf{ no} \text{ cumple la propiedad} \end{cases}$
- $f(x)$ es monótona $\star$

$$val = 22 \quad f(x) = \begin{cases} 1 & \text{si } x \geq val \\ 0 & \text{si } x < val \end{cases}$$

1	4	5	8	11	19	21	22	25	29	31	37	42
0	0	0	0	0	0	0	1	1	1	1	1	1

- Podemos buscar el primer 1 o el último 0

Forma general

```
int lo = 0, hi = n-1;
int res = -1;
while (lo <= hi) {
  int mid = (lo + hi) / 2;
  if (f(mid) == 1) {
    res = mid; // mejor respuesta hasta ahora
    hi = mid - 1;
  }
  else { // f(mid) == 0
    lo = mid + 1;
  }
}
// Primer 1 esta en la posicion res
```

Forma general

$val = 25$

1	4	5	8	11	19	21	22	25	29	31	37	42
lo						m						hi

Forma general

$val = 25$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo		m			hi
									res			

Forma general

$val = 25$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	lo	hi	×	×	×	×
							m					
									res			

Forma general

$val = 25$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	×	lo	×	×	×	×
								hi				
								m				
								res	~~res~~			

Forma general

$val = 25$

1	4	5	8	11	19	21	22	25	29	31	37	42
×	×	×	×	×	×	×	×	×	×	×	×	×
							hi	lo				
								res	~~res~~			

Problema de ejemplo

Tenemos un arreglo a que contiene n enteros.

Se deben responder q consultas de la forma: Dados dos valores l y r, ¿cuantos elementos del arreglo a están en el rango $[l, r]$?

Problema de ejemplo

Tenemos un arreglo a que contiene n enteros.

Se deben responder q consultas de la forma: Dados dos valores l y r, ¿cuantos elementos del arreglo a están en el rango $[l, r]$?

- No hay ninguna restriccion sobre el orden de los elementos. Podemos ordenar el arreglo para encontrar una función monótona

Problema de ejemplo

Tenemos un arreglo a que contiene n enteros.

Se deben responder q consultas de la forma: Dados dos valores l y r, ¿cuantos elementos del arreglo a están en el rango $[l, r]$?

- No hay ninguna restriccion sobre el orden de los elementos. Podemos ordenar el arreglo para encontrar una función monótona
- Para encontrar limite izquierdo, buscamos el primer elemento mayor o igual que l
- Función monótona $0, 0, ..., 0, 0, 0, 1, 1, 1, ...$

Problema de ejemplo

Tenemos un arreglo a que contiene n enteros.

Se deben responder q consultas de la forma: Dados dos valores l y r, ¿cuantos elementos del arreglo a están en el rango $[l, r]$?

- No hay ninguna restriccion sobre el orden de los elementos. Podemos ordenar el arreglo para encontrar una función monótona
- Para encontrar limite izquierdo, buscamos el primer elemento mayor o igual que l
- Función monótona $0, 0, ..., 0, 0, 0, 1, 1, 1, ...$
- Para encontrar el limite derecho, buscamos el ultimo elemento menor o igual que r
- Función monótona $1, 1, ..., 1, 1, 1, 0, 0, 0, ...$

Problema de ejemplo

Tenemos n piezas rectangulares de tamaño $a \times b$. ¿Cuál es el tamaño más pequeño que puede tener el lado de un cuadrado que contenga a todas las piezas?

Las piezas no se pueden rotar ni superponer. El tamaño del cuadrado debe ser entero.

Problema de ejemplo

Tenemos n piezas rectangulares de tamaño $a \times b$. ¿Cuál es el tamaño más pequeño que puede tener el lado de un cuadrado que contenga a todas las piezas?

Las piezas no se pueden rotar ni superponer. El tamaño del cuadrado debe ser entero.

- Si las n piezas entran en un cuadrado de lado x, entrarán en uno de lado $x + 1$
- Si las n piezas no entran en un cuadrado de lado x, no entrarán en uno de lado $x - 1$

Problema de ejemplo

Tenemos n piezas rectangulares de tamaño $a \times b$. ¿Cuál es el tamaño más pequeño que puede tener el lado de un cuadrado que contenga a todas las piezas?

Las piezas no se pueden rotar ni superponer. El tamaño del cuadrado debe ser entero.

- Si las n piezas entran en un cuadrado de lado x, entrarán en uno de lado $x + 1$
- Si las n piezas no entran en un cuadrado de lado x, no entrarán en uno de lado $x - 1$
- Función monótona $0, 0, ..., 0, 0, 0, 1, 1, 1, ...$

Problema de ejemplo

Tenemos n piezas rectangulares de tamaño $a \times b$. ¿Cuál es el tamaño más pequeño que puede tener el lado de un cuadrado que contenga a todas las piezas?

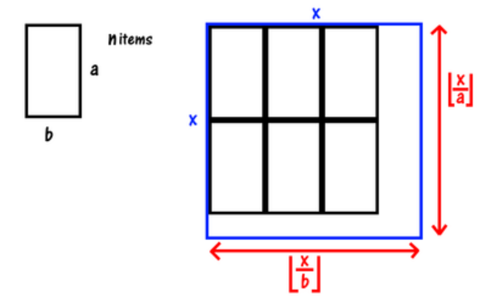
Las piezas no se pueden rotar ni superponer. El tamaño del cuadrado debe ser entero.

- Si las n piezas entran en un cuadrado de lado x, entrarán en uno de lado $x + 1$
- Si las n piezas no entran en un cuadrado de lado x, no entrarán en uno de lado $x - 1$
- Función monótona $0, 0, ..., 0, 0, 0, 1, 1, 1, ...$
- A esta forma de aplicar búsqueda binaria se le llama **binary search the answer**.

Problema de ejemplo

Un cuadrado de $x \cdot x$ puede:

- acomodar $\lfloor \frac{x}{a} \rfloor$ piezas de un lado
- acomodar $\lfloor \frac{x}{b} \rfloor$ piezas del otro lado



$$f(x) = \begin{cases} 1 & \text{si } \lfloor \frac{x}{a} \rfloor \cdot \lfloor \frac{x}{b} \rfloor \geq n \\ 0 & \text{si } \lfloor \frac{x}{a} \rfloor \cdot \lfloor \frac{x}{b} \rfloor < n \end{cases}$$

Problema de ejemplo

Tenemos n cuerdas, cada una de largo l_i. Necesitamos k cuerdas del mismo tamaño.

¿Cual es el largo mas grande que pueden tener las k cuerdas si podemos cortar las cuerdas originales de la forma que queramos?

La respuesta se considera correcta si el error relativo o absoluto no es mayor a 10^{-6}

Problema de ejemplo

Tenemos n cuerdas, cada una de largo l_i. Necesitamos k cuerdas del mismo tamaño.

¿Cual es el largo mas grande que pueden tener las k cuerdas si podemos cortar las cuerdas originales de la forma que queramos?

La respuesta se considera correcta si el error relativo o absoluto no es mayor a 10^{-6}

- Imaginemos que x es la respuesta optima

Problema de ejemplo

Tenemos n cuerdas, cada una de largo l_i. Necesitamos k cuerdas del mismo tamaño.

¿Cual es el largo mas grande que pueden tener las k cuerdas si podemos cortar las cuerdas originales de la forma que queramos?

La respuesta se considera correcta si el error relativo o absoluto no es mayor a 10^{-6}

- Imaginemos que x es la respuesta optima
- Podremos obtener k o mas cuerdas si las cortamos de largo **menor** que x
- Es imposible obtener k o mas cuerdas si las cortamos de largo **mayor** que x

Problema de ejemplo

Tenemos n cuerdas, cada una de largo l_i. Necesitamos k cuerdas del mismo tamaño.

¿Cual es el largo mas grande que pueden tener las k cuerdas si podemos cortar las cuerdas originales de la forma que queramos?

La respuesta se considera correcta si el error relativo o absoluto no es mayor a 10^{-6}

- Imaginemos que x es la respuesta optima
- Podremos obtener k o mas cuerdas si las cortamos de largo **menor** que x
- Es imposible obtener k o mas cuerdas si las cortamos de largo **mayor** que x
- La funcion es monótona $..., 1, 1, 0, 0, ...$ sobre un rango **continuo**

Problema de ejemplo

Input de ejemplo	Output de ejemplo
4 11 802 743 457 539	200.5

- Se pueden obtener al menos 11 cuerdas de largo 200.5

Problema de ejemplo

Input de ejemplo	Output de ejemplo

`4 11`
`802`
`743`
`457`
`539`
	`200.5`

- Se pueden obtener al menos 11 cuerdas de largo 200.5
- Tambien se pueden obtener al menos 11 cuerdas de largo 200.4, 200.45, 200.499999, etc.

Problema de ejemplo

Input de ejemplo	Output de ejemplo
4 11 802 743 457 539	200.5

- Se pueden obtener al menos 11 cuerdas de largo 200.5
- Tambien se pueden obtener al menos 11 cuerdas de largo 200.4, 200.45, 200.499999, etc.
- No es necesario que la respuesta sea exacta, pero debe ser un valor muy cercano.

Problema de ejemplo

```
... // leer n, k, arreglo l
double lo = 0, hi = cuerdaMasLarga;
double res = -1;
while (lo <= hi) {
  double mid = (lo + hi) / 2;
  int cuerdas = 0;
  for (int i = 0; i < n; i++) {
    cuerdas += floor(l[i] / mid);
  }
  if (cuerdas >= k) {
    res = mid;
    lo = mid + ???;
  }
  else {
    hi = mid - ???;
  }
}
```

Problema de ejemplo

```
... // leer n, k, arreglo l
double lo = 0, hi = cuerdaMasLarga;
for (int aux = 0; aux < 50; aux++) {
  double mid = (lo + hi) / 2;
  int cuerdas = 0;
  for (int i = 0; i < n; i++) {
    cuerdas += floor(l[i] / mid);
  }
  if (cuerdas >= k) lo = mid;
  else hi = mid;
}
```

Problema de ejemplo

```
... // leer n, k, arreglo l
double lo = 0, hi = cuerdaMasLarga;
for (int aux = 0; aux < 50; aux++) {
  double mid = (lo + hi) / 2;
  int cuerdas = 0;
  for (int i = 0; i < n; i++) {
    cuerdas += floor(l[i] / mid);
  }
  if (cuerdas >= k) lo = mid;
  else hi = mid;
}
```

- Cada iteracion reduce el rango de posibilidades a la mitad

Problema de ejemplo

```
... // leer n, k, arreglo l
double lo = 0, hi = cuerdaMasLarga;
for (int aux = 0; aux < 50; aux++) {
  double mid = (lo + hi) / 2;
  int cuerdas = 0;
  for (int i = 0; i < n; i++) {
    cuerdas += floor(l[i] / mid);
  }
  if (cuerdas >= k) lo = mid;
  else hi = mid;
}
```

- Cada iteracion reduce el rango de posibilidades a la mitad
- Luego de 50 iteraciones, el rango sera $\frac{maxCuerda}{2^{50}}$

Problema de ejemplo

```
... // leer n, k, arreglo l
double lo = 0, hi = cuerdaMasLarga;
for (int aux = 0; aux < 50; aux++) {
  double mid = (lo + hi) / 2;
  int cuerdas = 0;
  for (int i = 0; i < n; i++) {
    cuerdas += floor(l[i] / mid);
  }
  if (cuerdas >= k) lo = mid;
  else hi = mid;
}
```

- Cada iteracion reduce el rango de posibilidades a la mitad
- Luego de 50 iteraciones, el rango sera $\frac{maxCuerda}{2^{50}}$
- Probablemente suficientemente preciso