

Algoritmos y estructuras de datos

Ordenamiento por mezcla

Francisco Javier Zaragoza Martínez

Universidad Autónoma Metropolitana Unidad Azcapotzalco
Departamento de Sistemas

22 de mayo de 2015

Problema abstracto

Dado un arreglo A con n elementos, se desea reacomodar sus elementos de modo que $A_0 \leq A_1 \leq \dots \leq A_{n-1}$.

Problema concreto

Dado un arreglo `int a[MAX]` con `int n` elementos en las posiciones `a[0]`, `...`, `a[n-1]` se desea reacomodar sus elementos de modo que `a[0] <= a[1] <= ... <= a[n-1]`.

Un problema, mil soluciones

El problema de ordenamiento interno es uno de los más estudiados en la computación y para el que se conocen muchos algoritmos. Algunos son:

- Burbuja^a.
- Inserción directa.
- Selección directa.
- Quicksort.
- Ordenamiento por mezcla.
- Ordenamiento por montículo.

^aEste algoritmo se suele estudiar en un curso introductorio

Ordenamiento interno

Un problema, mil soluciones

El problema de ordenamiento interno es uno de los más estudiados en la computación y para el que se conocen muchos algoritmos. Algunos son:

- Burbuja^a.
- Inserción directa.
- Selección directa.
- Quicksort.
- Ordenamiento por mezcla.
- Ordenamiento por montículo.

^aEste algoritmo se suele estudiar en un curso introductorio

Comparaciones

Todos estos algoritmos **comparan** los datos en el arreglo.
Existen algoritmos que **no comparan** los datos en el arreglo.

Inserción directa

Ejemplo

3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	9	6	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	9	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	6	9	5	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	9	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	6	9	5	3	5	8	9	7	9	3
1	1	2	3	4	5	5	6	9	3	5	8	9	7	9	3

Inserción directa

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	3	4	1	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	3	4	5	9	2	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	9	6	5	3	5	8	9	7	9	3
1	1	2	3	4	5	6	9	5	3	5	8	9	7	9	3
1	1	2	3	4	5	5	6	9	3	5	8	9	7	9	3
1	1	2	3	3	4	5	5	6	9	5	8	9	7	9	3

Idea más general

Tenemos un arreglo ordenado $a[0] \leq a[1] \leq \dots \leq a[j-1]$ en el que queremos insertar $a[j]$. Esto se puede hacer con una modificación al código de inserción de un elemento en un conjunto ordenado.

```
void inserta(int j, int a[])
{
    int i;
    int t = a[j];

    for (i = j-1; i >= 0 && a[i] > t; i--)
        a[i+1] = a[i];
    a[i+1] = t;
}
```


Iterativo

```
void insercion(int n, int a[])
{
    int j;

    for (j = 1; j < n; j++)
        inserta(j, a);
}
```

Iterativo

```
void insercion(int n, int a[])
{
    int j;

    for (j = 1; j < n; j++)
        inserta(j, a);
}
```

Comparaciones

La llamada a `inserta(j, a)` hace **a lo mucho** j comparaciones. En total se hacen **a lo mucho** $1 + 2 + \dots + (n - 1) = \frac{1}{2}n(n - 1)$ comparaciones.

Recursivo

```
void insercion(int n, int a[])  
{  
    if (n > 1) {  
        insercion(n-1, a);  
        inserta(n-1, a);  
    }  
}
```

Recursivo

```
void insercion(int n, int a[])
{
    if (n > 1) {
        insercion(n-1, a);
        inserta(n-1, a);
    }
}
```

Comparaciones

Si la llamada a `insercion(n,a)` hace $T(n)$ comparaciones, entonces $T(1) = 0$ y $T(n) \leq T(n-1) + (n-1)$ si $n > 1$. En total se hacen $\leq (n-1) + \dots + 2 + 1 = \frac{1}{2}n(n-1)$ comparaciones.

Ejercicios

- 1 Reescribe `inserta` con apuntadores.
- 2 Reescribe `insercion` para que ordene decrecientemente.
- 3 Reescribe `insercion` para que inicie con el último elemento.
- 4 Reescribe `insercion` como dos ciclos anidados.
- 5 ¿Cuántas veces se lee del arreglo `a`?
- 6 ¿Cuántas veces se escribe en el arreglo `a`?

Ordenamiento por mezcla

Ejemplo

3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		i	i	?	?										

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		i	i	?	?										
1	1	2	3	4	5	6	9								

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		i	i	?	?										
1	1	2	3	4	5	6	9								
								5	3	5	8	9	7	9	3

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		¿	¿	?	?										
1	1	2	3	4	5	6	9								
								5	3	5	8	9	7	9	3
										¿	¿	?	?		

Ordenamiento por mezcla

Ejemplo

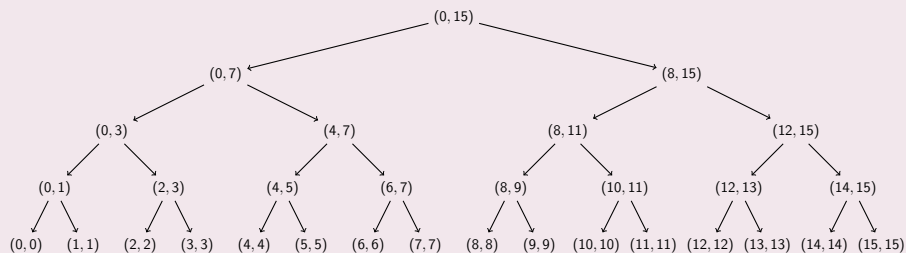
3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		¿	¿	?	?										
1	1	2	3	4	5	6	9								
								5	3	5	8	9	7	9	3
										¿	¿	?	?		
								3	3	5	5	7	8	9	9

Ordenamiento por mezcla

Ejemplo

3	1	4	1	5	9	2	6	5	3	5	8	9	7	9	3
3	1	4	1	5	9	2	6								
		¿	¿	?	?										
1	1	2	3	4	5	6	9								
								5	3	5	8	9	7	9	3
										¿	¿	?	?		
								3	3	5	5	7	8	9	9
1	1	2	3	3	3	4	5	5	5	6	7	8	9	9	9

Árbol de recursión



Unión de dos vectores ordenados

Esto lo podemos hacer con una modificación del código que usamos para la unión de dos conjuntos almacenados en vectores ordenados, pero ahora debemos permitir las repeticiones. Aquí `int b[]` es un arreglo **adicional**.

```
void une(int i, int m, int j, int a[])
{
    int p=i, q=m+1, r=i;

    while (p <= m && q <= j)
        b[r++] = (a[p] <= a[q]) ? a[p++] : a[q++];
    while (p <= m)
        b[r++] = a[p++];
    for (p = i; p < r; p++)
        a[p] = b[p];
}
```

Recursivo

Esta función deberá llamarse como `mezcla(0, n-1, a)`.

```
void mezcla(int i, int j, int a[])
{
    int m;

    if (i < j) {                /* dos o mas elementos */
        m = (i+j)/2;           /* calcula punto medio */
        mezcla(i, m, a);       /* ordena primera mitad */
        mezcla(m+1, j, a);     /* ordena segunda mitad */
        une(i, m, j, a);       /* une las dos mitades */
    }
}
```


Ordenamiento por mezcla

Comparaciones de la unión

Con un vector de tamaño n , `une` hace $\leq n - 1$ comparaciones.

Ordenamiento por mezcla

Comparaciones de la unión

Con un vector de tamaño n , `une` hace $\leq n - 1$ comparaciones.

Comparaciones de ordenamiento por mezcla

Si $T(n)$ es el número de comparaciones que hace `mezcla` con un vector de tamaño n , entonces $T(1) = 0$ y $T(n) \leq 2T(n/2) + (n - 1)$.

Ordenamiento por mezcla

Comparaciones de la unión

Con un vector de tamaño n , `une` hace $\leq n - 1$ comparaciones.

Comparaciones de ordenamiento por mezcla

Si $T(n)$ es el número de comparaciones que hace `mezcla` con un vector de tamaño n , entonces $T(1) = 0$ y $T(n) \leq 2T(n/2) + (n - 1)$.

Calculando $T(n)$ para las potencias de 2

$$\begin{aligned}T(2) &\leq 2T(1) + (2 - 1) = 1 \\T(4) &\leq 2T(2) + (4 - 1) = 5 \\T(8) &\leq 2T(4) + (8 - 1) = 17 \\T(16) &\leq 2T(8) + (16 - 1) = 49.\end{aligned}$$

Calculando $T(n)$ para las potencias de 2

Demostremos por inducción en $k \geq 0$ que $T(2^k) \leq (k - 1)2^k + 1$.

- 1 Para $k = 0$ se cumple pues $0 = T(2^0) \leq (0 - 1)2^0 + 1 = 0$.
- 2 Supongamos que para alguna $k \geq 0$ se cumple $T(2^k) \leq (k - 1)2^k + 1$.
- 3 Entonces para $k + 1$ tenemos que:

$$\begin{aligned} T(2^{k+1}) &\leq 2T(2^k) + 2^{k+1} - 1 \text{ (fórmula recursiva)} \\ &\leq 2[(k - 1)2^k + 1] + 2^{k+1} - 1 \text{ (hipótesis)} \\ &= (k - 1)2^{k+1} + 2 + 2^{k+1} - 1 \\ &= k2^{k+1} + 1. \end{aligned}$$

Esto es lo que queríamos demostrar.

Es decir, $T(n) \approx n \log_2 n$.

Ordenamiento por mezcla

Comparación

Propiedad	Burbuja	Inserción	Mezcla
Memoria	n	n	$2n$
Comparaciones	$\frac{1}{2}n^2$	$\frac{1}{2}n^2$	$n \log_2 n$

Ordenamiento por mezcla

Comparación

Propiedad	Burbuja	Inserción	Mezcla
Memoria	n	n	$2n$
Comparaciones	$\frac{1}{2}n^2$	$\frac{1}{2}n^2$	$n \log_2 n$

Otra comparación

Tiempo necesario para ordenar n datos si cada comparación tarda 1 ns.

n	Burbuja	Inserción	Mezcla
10^3	0.5 ms	0.5 ms	0.01 ms
10^4	50 ms	50 ms	0.1 ms
10^5	5 s	5 s	1 ms
10^6	500 s	500 s	20 ms

Ejercicios

- 1 Escribe una función recursiva `int minimo(int i, int j, int a[])` que encuentre el mínimo de (a_i, \dots, a_j) usando la siguiente idea: Si $i = j$ entonces el mínimo es a_i . De lo contrario, calcula $m = \lfloor \frac{i+j}{2} \rfloor$ y resuelve recursivamente para (a_i, \dots, a_m) y (a_{m+1}, \dots, a_j) .
¿Qué debes hacer con las respuestas de las llamadas recursivas?
- 2 Vuelve a hacerlo para ahora encontrar el máximo de (a_i, \dots, a_j) .
- 3 Vuelve a hacerlo para ahora encontrar **simultáneamente** el mínimo y el máximo de (a_i, \dots, a_j) .