

Quicksort

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int contint=0, ciclo=0;

void quicksort(int a[], int primero, int ultimo)
{
    int pivote, i, j, central, tmp;
    central = (primero + ultimo)/2;
    pivote = a[central];
    i = primero;
    j = ultimo;
    do {
        while (a[i] < pivote) i++;
        while (a[j] > pivote) j--;
        if (i<=j)
        {
            tmp = a[i];
            a[i] = a[j];
            a[j] = tmp; /* intercambia a[i] con a[j] */
            i++;
            j--;
            printf("Intercambio Nro. %d\n", contint++);
        }
        ciclo++;
    }while (i <= j);
    if (primero < j) quicksort(a, primero, j);/* mismo proceso con
    sublista izqda */
    if (i < ultimo) quicksort(a, i, ultimo); /* mismo proceso con
    sublista drcha */

} //fin do-while

```

```

main()
{
    const int b=10;

    int A[b]={8,1,4,9,6,3,5,2,7,0},c;

```

quicksort (A, 0, 9); // llamado recursivo

```

printf("\nValores ordenados: \n");

for(c=0;c<b;c++){ printf("%d, ",A[c]); }
printf("\n");

system("PAUSE");
return 0;
}

```

Comenzamos...

```
void quicksort(int a[], int primero, int ultimo)
{
    int pivote, i, j, central, tmp;
    central = (primero + ultimo)/2;
    pivote = a[central];
    i = primero;
    j = ultimo;
```

.....

Valor	8	1	4	9	6	3	5	2	7	0
Posición	0	1	2	3	4	5	6	7	8	9

```
Valor de central -(primero + ultimo)/2- indica posicion del pivote: 4
Valor de pivote -a[central]- indica valor del pivote: 6
Valor de primero -0- se asigna a i
Valor de ultimo -9- se asigna a j
```

Al comenzar el ciclo $i=0$, $j=9$.

Valor	8	1	4	9	6	3	5	2	7	0
Posición	0 i	1	2	3	4	5	6	7	8	9 j

```
.....
do {
  while ( $a[i] < pivote$ ) i++;
  while ( $a[j] > pivote$ ) j--;
  if (i<=j)
  {
    tmp = a[i];
    a[i] = a[j];
    a[j] = tmp; /* intercambia a[i] con a[j] */
    i++;
    j--;
    printf("Intercambio Nro. %d\n", contint++);
  }
  ciclo++;
}while (i <= j);
```

Falso $\rightarrow i=0$

Falso $\rightarrow j=9$

```
Comienzo del ciclo do while = pivote:6, i:0, j:9
```

```
Dentro del ciclo do while = pivote:6, i:0, j:9
```

```
Dentro del intercambio, variable i:1
```

```
Dentro del intercambio, variable j:8
```

```
Intercambio Nro. 1
```

```
0-1-4-9-6-3-5-2-7-8-
```

Valor	0	1	4	9	6	3	5	2	7	8
Posición	0	1 i	2	3	4	5	6	7	8 j	9

.....

do {

while (**a[i] < pivote**) i++;

Verdadero → i=3, se incrementa

while (**a[j] > pivote**) j--;

Verdadero → j=7, se decrementa

Valor	0	1	4	9	6	3	5	2	7	8
Posición	0	1	2	3 i	4	5	6	7 j	8	9

if (i<=j)

{

tmp = a[i];

a[i] = a[j];

a[j] = tmp; /* intercambia a[i] con a[j] */

i++;

j--;

printf("Intercambio Nro. %d\n", contint++);

}

ciclo++;

}while (i <= j);

Valor	0	1	4	2	6	3	5	9	7	8
Posición	0	1	2	3	4 i	5	6 j	7	8	9

```
Dentro del ciclo do while = pivote:6, i:1, j:8
while->variable i (U:se incrementa) 2
while->variable i (U:se incrementa) 3
while->variable j (U:se decrementa) 7
Dentro del intercambio, variable i:4
Dentro del intercambio, variable j:6
Intercambio Nro. 2
0-1-4-2-6-3-5-9-7-8-
```

Valor	0	1	4	2	6	3	5	9	7	8
Posición	0	1	2	3	4 i	5	6 j	7	8	9

.....
do {

while (**a[i] < pivote**) i++;

Falso → i=4, no se incrementa

while (**a[j] > pivote**) j--;

Falso → j=6, no se decrementa

if (i<=j)

{

tmp = a[i];

a[i] = a[j];

a[j] = tmp; /* intercambia a[i] con a[j] */

i++;

j--;

printf("Intercambio Nro. %d\n", contint++);

}

ciclo++;

}while (i <= j);

```
Dentro del ciclo do while = pivote:6, i:4, j:6
Dentro del intercambio, variable i:5
Dentro del intercambio, variable j:5
Intercambio Nro. 3
0-1-4-2-5-3-6-9-7-8-
```

Valor	0	1	4	2	5	3	6	9	7	8
Posición	0	1	2	3	4	5 i j	6	7	8	9

Valor	0	1	4	2	5	3	6	9	7	8
Posición	0	1	2	3	4	5	6	7	8	9

```

.....
do {
    while (a[i] < pivote) i++;
    while (a[j] > pivote) j--;

    if (i<=j)
    {
        tmp = a[i];
        a[i] = a[j];
        a[j] = tmp; /* intercambia a[i] con a[j] */
        i++;
        j--;
        printf("Intercambio Nro. %d\n", contint++);
    }
    ciclo++;
}while (i <= j);

```

Valor	0	1	4	2	5	3	6	9	7	8
Posición	0	1	2	3	4	5	6	7	8	9

```

Dentro del ciclo do while = pivote:6, i:5, j:5
while->variable i <U:se incrementa> 6
0-1-4-2-5-3-6-9-7-8-

```

```

Si i es > a j, termino el do-while, hubo 3 intercambio/s, pivote:6, i:6, j:5

```

Valor	0	1	4	2	5	3	6	9	7	8
Posición	0	1	2	3	4	5 j	6 i	7	8	9

```
if (primero < j) quicksort(a, primero, j);/* mismo proceso con sublista izqda */
if (i < ultimo) quicksort(a, i, ultimo); /* mismo proceso con sublista drcha */
```

```
Llamado recursivo comparando primero:0 con j:5
Valor de central -(primero + ultimo)/2- indica posicion del pivote: 2
Valor de pivote -a[centrall]- indica valor del pivote: 4
Valor de primero -0- se asigna a i
Valor de ultimo -5- se asigna a j
```

Valor	0	1	4	2	5	3	6	9	7	8
Posición	0 i	1	2	3	4	5 j	6	7	8	9

```
Comienzo del ciclo do while = pivote:4, i:0, j:5
Dentro del ciclo do while = pivote:4, i:0, j:5
while->variable i (0:se incrementa) 1
while->variable i (0:se incrementa) 2
Dentro del intercambio, variable i:3
Dentro del intercambio, variable j:4
Intercambio Nro. 4
0-1-3-2-5-4-6-9-7-8-
```


Valor	0	1	3	2	5	4	6	9	7	8
Posición	0	1	2	3 i	4	5 j	6	7	8	9

```

.....
do {
    while (a[i] < pivote) i++;
    while (a[j] > pivote) j--;

    if (i<=j)
    {
        tmp = a[i];
        a[i] = a[j];
        a[j] = tmp; /* intercambia a[i] con a[j] */
        i++;
        j--;
        printf("Intercambio Nro. %d\n", contint++);
    }
    ciclo++;
}while (i <= j);

```

Valor	0	1	3	2	5	4	6	9	7	8
Posición	0	1	2	3 j	4 i	5	6	7	8	9

```

Dentro del ciclo do while = pivote:4, i:3, j:4
while->variable i (U:se incrementa) 4
while->variable j (U:se decrementa) 3
0-1-3-2-5-4-6-9-7-8-

```

Si i es > a j, termino el do-while, hubo 4 intercambio/s, pivote:4, i:4, j:3

```
Llamado recursivo comparando primero:0 con j:3
```

```
Valor de central -(primero + ultimo)/2- indica posicion del pivote: 1
```

```
Valor de pivote -alcentrall- indica valor del pivote: 1
```

```
Valor de primero -0- se asigna a i
```

```
Valor de ultimo -3- se asigna a j
```

Valor	0	1	3	2	5	4	6	9	7	8
Posición	0 i	1	2	3 j	4	5	6	7	8	9

```
.....
do {
    while (a[i] < pivote) i++;
    while (a[j] > pivote) j--;

    if (i <= j)
    {
        tmp = a[i];
        a[i] = a[j];
        a[j] = tmp; /* intercambia a[i] con a[j] */
        i++;
        j--;
        printf("Intercambio Nro. %d\n", contint++);
    }
    ciclo++;
}while (i <= j);
```

```
Comienzo del ciclo do while = pivote:1, i:0, j:3
```

```
Dentro del ciclo do while = pivote:1, i:0, j:3
```

```
while->variable i <U:se incrementa> 1
```

```
while->variable j <U:se decrementa> 2
```

```
while->variable j <U:se decrementa> 1
```

```
Dentro del intercambio, variable i:2
```

```
Dentro del intercambio, variable j:0
```

```
Intercambio Nro. 5
```

```
0-1-3-2-5-4-6-9-7-8-
```

```
Si i es > a j, termino el do-while, hubo 5 intercambio/s, pivote:1, i:2, j:0
```

```
Llamado recursivo comparando primero:0 con j:0
```

```
Llamado recursivo comparando ultimo:3 con j:0
```

```
Valor de central -(primero + ultimo)/2- indica posicion del pivote: 2
```

```
Valor de pivote -a[central]- indica valor del pivote: 3
```

```
Valor de primero -2- se asigna a i
```

```
Valor de ultimo -3- se asigna a j
```

Valor	0	1	3	2	5	4	6	9	7	8
Posición	0	1	2 i	3 j	4	5	6	7	8	9

```
.....  
do {
```

```
    while (a[i] < pivote) i++;
```

```
    while (a[j] > pivote) j--;
```

```
    if (i<=j)
```

```
    {
```

```
        tmp = a[i];
```

```
        a[i] = a[j];
```

```
        a[j] = tmp; /* intercambia a[i] con a[j] */
```

```
        i++;
```

```
        j--;
```

```
        printf("Intercambio Nro. %d\n", contint++);
```

```
    }
```

```
    ciclo++;
```

```
}while (i <= j);
```

```
Comienzo del ciclo do while = pivote:3, i:2, j:3
```

```
Dentro del ciclo do while = pivote:3, i:2, j:3
```

```
Dentro del intercambio, variable i:3
```

```
Dentro del intercambio, variable j:2
```

```
Intercambio Nro. 6
```

```
0-1-2-3-5-4-6-9-7-8-
```

```
Si i es > a j, termino el do-while, hubo 6 intercambio/s, pivote:3, i:3, j:2
```

Fin