	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación

salas A y B

Profesor : Edgar Tista García

Asignatura: EDA 2

Grupo: 4

No. Práctica : 2

Integrante: Chagoya Gonzalez Leonardo

No. Equipo de Cómputo: N/A

No. de Lista o Brigada: N/A

Semestre: 2022-2

Fecha de entrega: 25/02/2022

Observaciones:

CALIFICACIÓN: _____

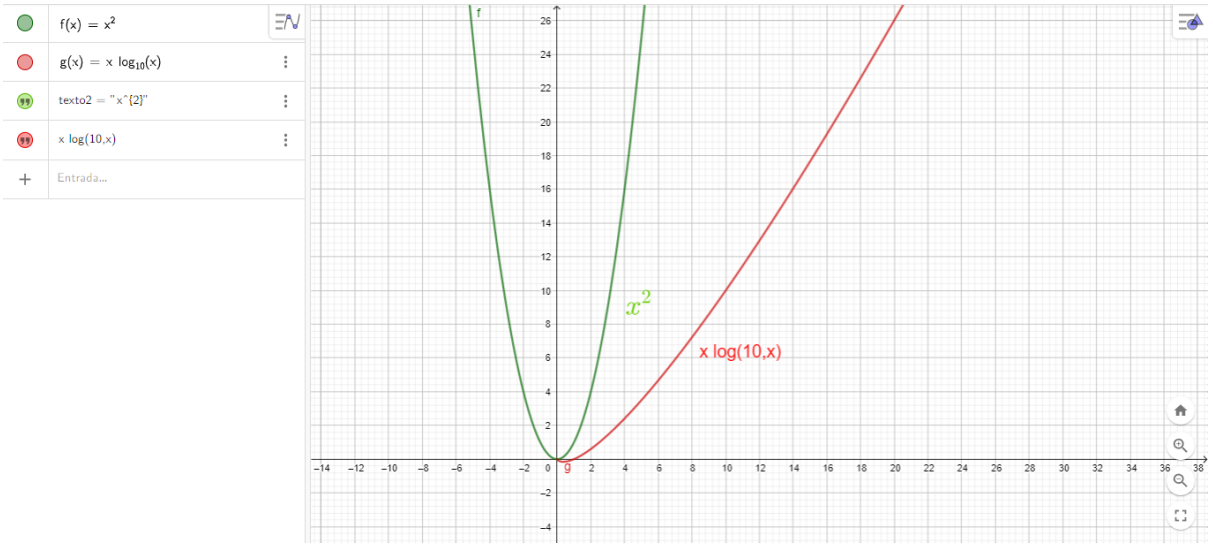
Objetivo: El estudiante identificará la estructura de los algoritmos de ordenamiento HeapSort, QuickSort y MergeSort

Objetivo de clase: El estudiante observará la importancia del orden de complejidad aplicado en algoritmos de ordenamiento, conocerá diferentes formas de implementar Quicksort y comenzará a realizar programas sencillos en el lenguaje Java

Ejercicio 1. Agregando Ordenamientos

a) Escribe un análisis general de los nuevos algoritmos proporcionados

Tanto HeapSort como QuickSort y MergeSort son mejores algoritmos hablando de tiempos de ejecución ya que estos tienen complejidad O(n log n) para todos sus casos -exceptuando el peor caso o casos cercanos al peor en QuickSort en donde su complejidad aumenta a $O(n^2)$- mientras que SelectionSort, BubbleSort e Insertion Sort presentan complejidad $O(n^2)$ - a menos que se encuentre el mejor caso y se utilice BubbleSort donde su complejidad disminuye a $O(n)$ – lo cual indica que mientras más grande sea nuestra lista a ordenar mucho más tiempo de ejecución se llevará en realizar aquellos algoritmos con complejidad $O(n^2)$ en comparación de aquellos que tienen complejidad O(n log n) los cuales mantendrán cierta proporción y son más cercanos a una función lineal.

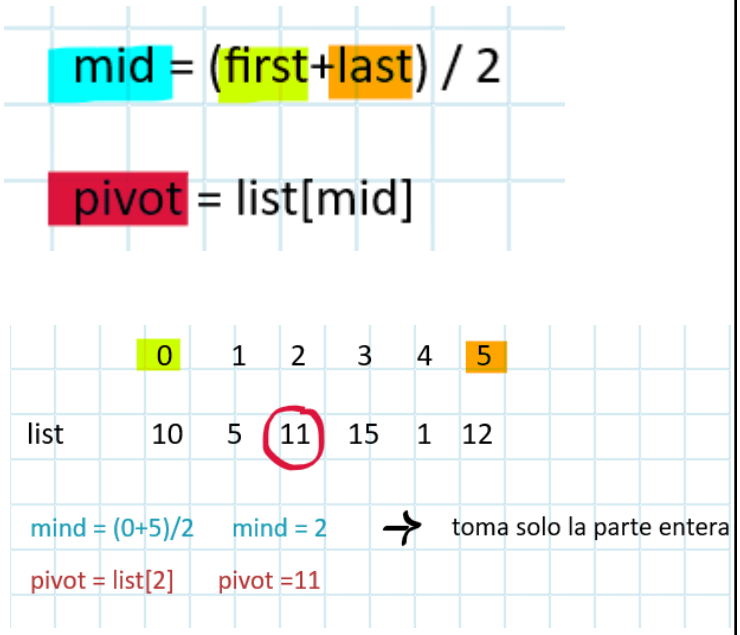
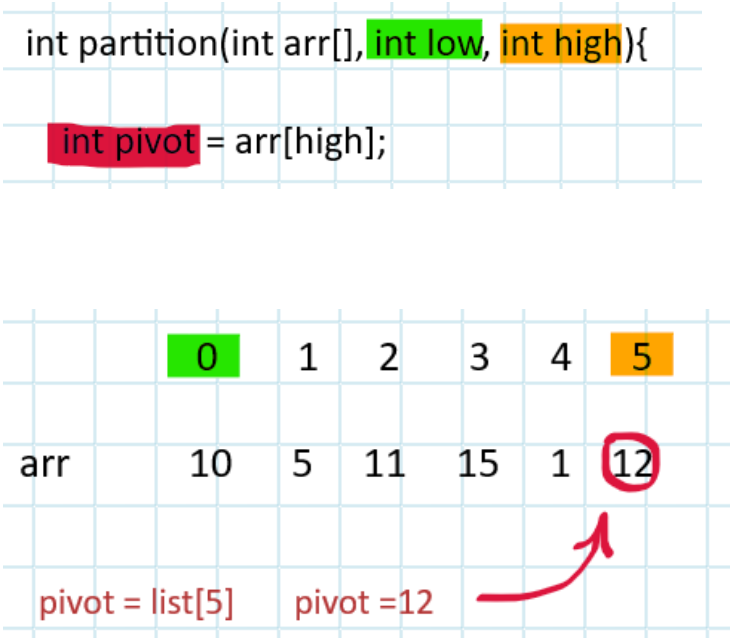


Comparativa de funciones n^2 y n log n

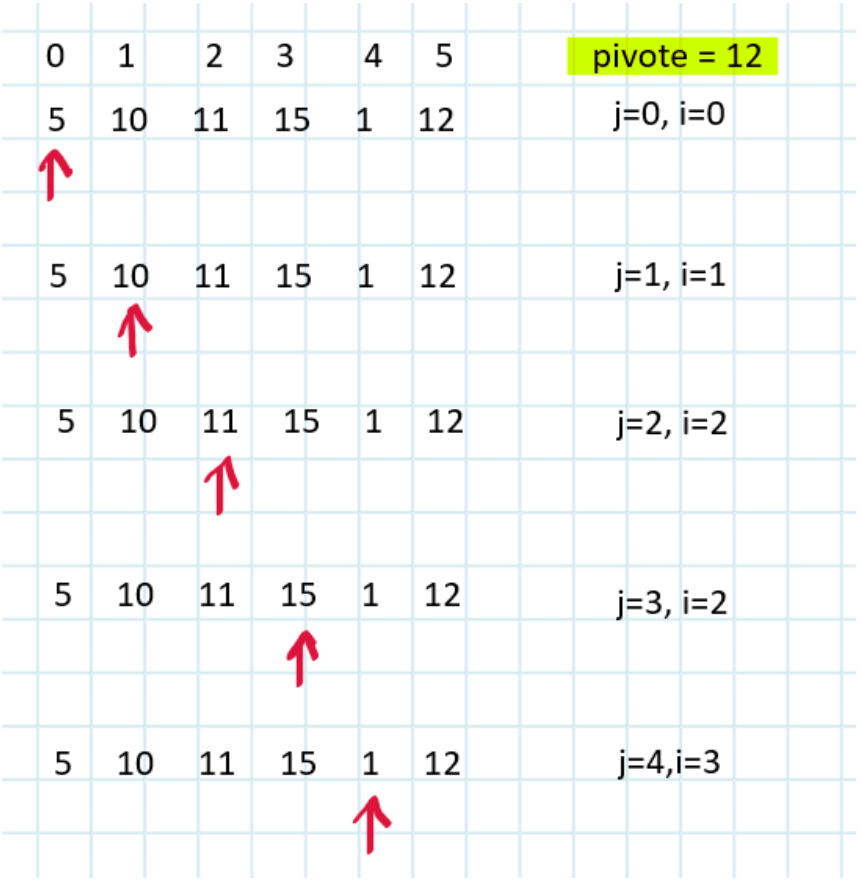
En el caso específico de HeapSort presenta una desventaja al utilizar la técnica de Top Down para convertir el arreglo a un Heap ya que este requiere memoria adicional, lo cual en arreglos de gran tamaño compromete el uso de este mismo.

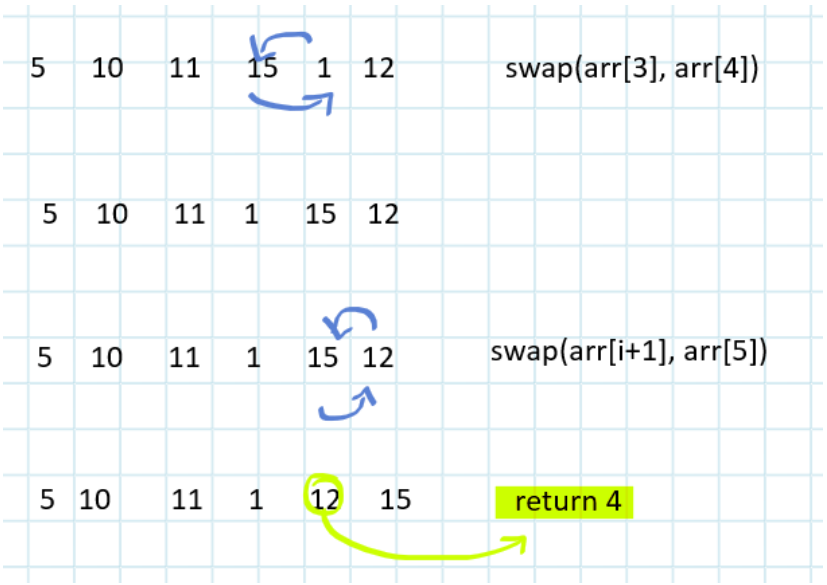
b) Revisa cuidadosamente el algoritmo de QuickSort e indica si se trata de alguna de las implementaciones vistas en clase o es una diferente.

No es ninguna de las implementaciones anteriormente vistas ya que el QuickSort que se proporciona en el código toma como pivote el último elemento del arreglo, el del código de los videos toma el valor que se encuentra enmedio y el que se observa en clase toma el primer elemento del arreglo.

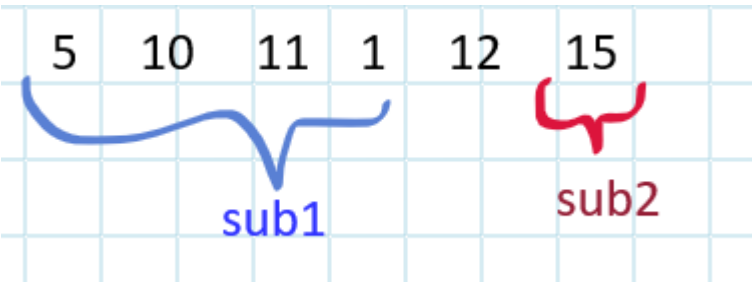
Algoritmo visto en clase	Algoritmo proporcionado en la práctica
	

Esto influye mucho pues en el algoritmo proporcionado en la práctica se concentra primero en toda la parte izquierda recorriendo el arreglo desde la posición *menor* hasta la posición del *pivote-1*,mediante la variable j,verificando si los elementos son menores que el pivote en dicho caso *i* aumentará su valor y se realizará un intercambio entre el elemento de la posición *i* con el elemento de la posición *j* en turno. Esto se realiza con el fin de que todo a la izquierda del elemento pivote sea menor





Al final se busca encontrar la posición del elemento pivote dentro del arreglo de tal forma que este ya esté ordenado y retorna dicho valor para que funcione como un elemento que partirá el arreglo en sub arreglos



Y dichos sub arreglos serán ordenados haciendo uso de recursividad, primero se enfocará en la parte izquierda para posteriormente ir con la derecha en cada sub arreglo generado.

c) En el caso de heapsort, describe las funciones asociando lo visto en los videos de teoría

Heapify- Se encarga de la lógica para conseguir los valores hijos e irlos comparando con el padre-Verificación de un Max-Heap - en caso de que los hijos sean mayores que el padre se realiza un intercambio, gracias a esto la función Heapify puede ser llamada desde build Heap para construir un Heap por medio de la técnica Bottom Up, pero si se llama desde Heapsort y previamente se cambia el valor de la raíz por el último elemento del Heap se realizará todo el proceso recursivo de eliminación de raíces.

Build Heap -Se encarga de la parte de iteración de los padres, comenzando por el último padre que tiene hijos para ir decrementando hasta llegar a la raíz

HeapSort- Construye un Max-Heap y una vez construido empieza a realizar la eliminación de raíces.

2.- MergeSort

El algoritmo fue llevado a C de manera exitosa

```c
void mergeSort(int *array,int low, int high) {

    int q;
    if(low<high) {
        q = (low+high)/2;
        mergeSort(array,low,q);
        mergeSort(array,q+1,high);
        merge(array,low,q,high);
    }
}
```

```c
void merge(int *array, int p, int q, int r) {

    int i,j,k;
    int list[r+1];
    k=0;
    i=p;
    j=q+1;

    while(i<=q && j<=r) {
        if(array[i]<array[j]) {
            list[k++] = array[i++];
        } else
            list[k++] = array[j++];
    }
```

```c
    while(i<=q) {
        list[k++] = array[i++];
    }
    while(j<=r) {

        list[k++]=array[j++];
    }

    printf("Sub arreglo 1 ");
    printSubArray(array,p,q);

    printf("Sub arreglo 2 ");
    printSubArray(array,q+1,r);

    printf("Mezcla: ");
    for(i=r; i>=p; i--) {
        array[i]=list[--k];
    }
    printSubArray(array,p,r);
    printf("\n");
```

En lo personal la dificultad que representó este algoritmo fue entender cómo iba dividiendo el arreglo por medio de los índices y la realización de la mezcla una vez que ya no se puede dividir más, esto en consecuencia de utilizar llamadas recursivas, por lo cual se debe de tomar en cuenta los datos del frame correspondiente. En cuanto al código la dificultad se presentó en que para el pseudocódigo tomaba list2 como un objeto de la clase list al cual en principio no se percibe cuáles son los datos iniciales, sin embargo dicho arreglo nos ayudará para realizar la mezcla de elementos de los sub arreglos este debe tener tantos elementos como los sub arreglos a analizar dicho dato lo da la variable *r +1* ya que r es el indice del ultimo elemento del sub arreglo.

3.- Verificando el funcionamiento

HEAPSORT

```
El arreglo es: 202 52 926 482 777 211 198 342 374 664 207 805 391 379 885 449 448 968 233 986

--------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   4

---HeapSort---
202 52 926 482 777 211 198 342 374 986 207 805 391 379 885 449 448 968 233 664
202 52 926 482 777 211 198 342 968 986 207 805 391 379 885 449 448 374 233 664
202 52 926 482 777 211 198 449 968 986 207 805 391 379 885 342 448 374 233 664
202 52 926 482 777 211 885 449 968 986 207 805 391 379 198 342 448 374 233 664
202 52 926 482 777 805 885 449 968 986 207 211 391 379 198 342 448 374 233 664
202 52 926 482 986 805 885 449 968 777 207 211 391 379 198 342 448 374 233 664
202 52 926 968 986 805 885 449 482 777 207 211 391 379 198 342 448 374 233 664
202 986 926 968 52 805 885 449 482 777 207 211 391 379 198 342 448 374 233 664
202 986 926 968 777 805 885 449 482 52 207 211 391 379 198 342 448 374 233 664
202 986 926 968 777 805 885 449 482 664 207 211 391 379 198 342 448 374 233 52
986 202 926 968 777 805 885 449 482 664 207 211 391 379 198 342 448 374 233 52
986 968 926 202 777 805 885 449 482 664 207 211 391 379 198 342 448 374 233 52
986 968 926 482 777 805 885 449 202 664 207 211 391 379 198 342 448 374 233 52
986 968 926 482 777 805 885 449 374 664 207 211 391 379 198 342 448 202 233 52
terminó de construir el heap
Iteracion HS:
52 968 926 482 777 805 885 449 374 664 207 211 391 379 198 342 448 202 233 986
968 52 926 482 777 805 885 449 374 664 207 211 391 379 198 342 448 202 233 986
968 777 926 482 52 805 885 449 374 664 207 211 391 379 198 342 448 202 233 986
968 777 926 482 664 805 885 449 374 52 207 211 391 379 198 342 448 202 233 986
Iteracion HS:
233 777 926 482 664 805 885 449 374 52 207 211 391 379 198 342 448 202 968 986
926 777 233 482 664 805 885 449 374 52 207 211 391 379 198 342 448 202 968 986
926 777 885 482 664 805 233 449 374 52 207 211 391 379 198 342 448 202 968 986
926 777 885 482 664 805 379 449 374 52 207 211 391 233 198 342 448 202 968 986
```

```
Iteracion HS:
211 374 448 233 342 391 379 198 202 52 207 449 482 664 777 805 885 926 968 986
448 374 211 233 342 391 379 198 202 52 207 449 482 664 777 805 885 926 968 986
448 374 391 233 342 211 379 198 202 52 207 449 482 664 777 805 885 926 968 986
Iteracion HS:
207 374 391 233 342 211 379 198 202 52 448 449 482 664 777 805 885 926 968 986
391 374 207 233 342 211 379 198 202 52 448 449 482 664 777 805 885 926 968 986
391 374 379 233 342 211 207 198 202 52 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
52 374 379 233 342 211 207 198 202 391 448 449 482 664 777 805 885 926 968 986
379 374 52 233 342 211 207 198 202 391 448 449 482 664 777 805 885 926 968 986
379 374 211 233 342 52 207 198 202 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
202 374 211 233 342 52 207 198 379 391 448 449 482 664 777 805 885 926 968 986
374 202 211 233 342 52 207 198 379 391 448 449 482 664 777 805 885 926 968 986
374 342 211 233 202 52 207 198 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
198 342 211 233 202 52 207 374 379 391 448 449 482 664 777 805 885 926 968 986
342 198 211 233 202 52 207 374 379 391 448 449 482 664 777 805 885 926 968 986
342 233 211 198 202 52 207 374 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
207 233 211 198 202 52 342 374 379 391 448 449 482 664 777 805 885 926 968 986
233 207 211 198 202 52 342 374 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
52 207 211 198 202 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
211 207 52 198 202 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
202 207 52 198 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
207 202 52 198 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
198 202 52 207 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
202 198 52 207 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
Iteracion HS:
52 198 202 207 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
```

El arreglo final ordenado es:

```
El arreglo ordenado es: 52 198 202 207 211 233 342 374 379 391 448 449 482 664 777 805 885 926 968 986
```

QuickSort

```
El arreglo es: 627 620 576 819 943 784 355 293 145 982 501 537 217 801 223 322 433 586 201 433

-------------Menu-----------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   5

---QuickSort---

El pivote es: 433
Una vez colocado el pivote en su posicion 355 293 145 217 223 322 433 201 433 982 501 537 819 801 943 784 627 586 620 576
Izquierda Sub array :  355 293 145 217 223 322 433 201

El pivote es: 201
Una vez colocado el pivote en su posicion 145 201 355 217 223 322 433 293
Izquierda Sub array :  145
Derecha Sub array :  355 217 223 322 433 293

El pivote es: 293
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 433 355
Izquierda Sub array :  217 223

El pivote es: 223
Una vez colocado el pivote en su posicion 145 201 217 223
Izquierda Sub array :  217
Derecha Sub array :
Derecha Sub array :  322 433 355

El pivote es: 355
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433
Izquierda Sub array :  322
Derecha Sub array :  433
Derecha Sub array :  982 501 537 819 801 943 784 627 586 620 576

El pivote es: 576
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 819 801 943 784 627 586 620 982
Izquierda Sub array :  501 537
```

```
El pivote es: 537
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537
Izquierda Sub array :  501
Derecha Sub array :
Derecha Sub array :  819 801 943 784 627 586 620 982

El pivote es: 982
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 819 801 943 784 627 586 620 982
Izquierda Sub array :  819 801 943 784 627 586 620

El pivote es: 620
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 586 620 943 784 627 819 801
Izquierda Sub array :  586
Derecha Sub array :  943 784 627 819 801

El pivote es: 801
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 586 620 784 627 801 819 943
Izquierda Sub array :  784 627

El pivote es: 627
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 586 620 627 784
Izquierda Sub array :
Derecha Sub array :  784
Derecha Sub array :  819 943

El pivote es: 943
Una vez colocado el pivote en su posicion 145 201 217 223 293 322 355 433 433 501 537 576 586 620 627 784 801 819 943
Izquierda Sub array :  819
Derecha Sub array :
Derecha Sub array :

El arreglo ordenado es: 145 201 217 223 293 322 355 433 433 501 537 576 586 620 627 784 801 819 943 982
```

MERGESORT

```
El arreglo es: 976 520 663 856 499 342 953 706 229 364 921 245 425 698 547 576 320 602 129 4

--------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   6

---MergeSort---
Sub arreglo 1 Sub array :  976
Sub arreglo 2 Sub array :  520
Mezcla: Sub array :  520 976

Sub arreglo 1 Sub array :  520 976
Sub arreglo 2 Sub array :  663
Mezcla: Sub array :  520 663 976

Sub arreglo 1 Sub array :  856
Sub arreglo 2 Sub array :  499
Mezcla: Sub array :  499 856

Sub arreglo 1 Sub array :  520 663 976
Sub arreglo 2 Sub array :  499 856
Mezcla: Sub array :  499 520 663 856 976

Sub arreglo 1 Sub array :  342
Sub arreglo 2 Sub array :  953
Mezcla: Sub array :  342 953

Sub arreglo 1 Sub array :  342 953
Sub arreglo 2 Sub array :  706
Mezcla: Sub array :  342 706 953

Sub arreglo 1 Sub array :  229
Sub arreglo 2 Sub array :  364
Mezcla: Sub array :  229 364
```

```
Sub arreglo 1 Sub array :  342 706 953
Sub arreglo 2 Sub array :  229 364
Mezcla: Sub array :  229 342 364 706 953

Sub arreglo 1 Sub array :  499 520 663 856 976
Sub arreglo 2 Sub array :  229 342 364 706 953
Mezcla: Sub array :  229 342 364 499 520 663 706 856 953 976

Sub arreglo 1 Sub array :  921
Sub arreglo 2 Sub array :  245
Mezcla: Sub array :  245 921

Sub arreglo 1 Sub array :  245 921
Sub arreglo 2 Sub array :  425
Mezcla: Sub array :  245 425 921

Sub arreglo 1 Sub array :  698
Sub arreglo 2 Sub array :  547
Mezcla: Sub array :  547 698

Sub arreglo 1 Sub array :  245 425 921
Sub arreglo 2 Sub array :  547 698
Mezcla: Sub array :  245 425 547 698 921

Sub arreglo 1 Sub array :  576
Sub arreglo 2 Sub array :  320
Mezcla: Sub array :  320 576

Sub arreglo 1 Sub array :  320 576
Sub arreglo 2 Sub array :  602
Mezcla: Sub array :  320 576 602

Sub arreglo 1 Sub array :  129
Sub arreglo 2 Sub array :  4
Mezcla: Sub array :  4 129

Sub arreglo 1 Sub array :  320 576 602
Sub arreglo 2 Sub array :  4 129
Mezcla: Sub array :  4 129 320 576 602

Sub arreglo 1 Sub array :  245 425 547 698 921
Sub arreglo 2 Sub array :  4 129 320 576 602
Mezcla: Sub array :  4 129 245 320 425 547 576 602 698 921
```

Con la última mezcla de sub arreglos obtenemos el arreglo ordenado

```
Sub arreglo 1 Sub array :  229 342 364 499 520 663 706 856 953 976
Sub arreglo 2 Sub array :  4 129 245 320 425 547 576 602 698 921
Mezcla: Sub array :  4 129 229 245 320 342 364 425 499 520 547 576 602 663 698 706 856 921 953 976


El arreglo ordenado es: 4 129 229 245 320 342 364 425 499 520 547 576 602 663 698 706 856 921 953 976
```

Como se pudo observar en cada uno de los algoritmos para arreglos de 20 elementos se comprueba el funcionamiento de estos, para un mayor entendimiento de cómo se realiza el proceso de ordenamiento se añadieron print´s a los algoritmos de merge y quick de tal forma que mostraran sus procedimientos.

Para el caso de merge empieza a realizar los print de los arreglos una vez que estos ya no tienen más de donde partir por medio de los índices -tamaño igual a 1- y empieza a realizar las mezclas

Para el caso de quicksort imprime el elemento que toma de pivote, como lo acomoda en el arreglo y después se analiza el sub arreglo de la izquierda en caso de que exista, en caso contrario analiza el sub arreglo de la derecha repitiendo el proceso hasta que ya no pueda partir más y empieza a unir arreglos.

Una vez analizados los algoritmos decidir dónde se colocarían los print´s para una mayor visibilidad del proceso fue más sencillo. Por lo que en este punto no se presentaron muchos inconvenientes.

4.- Complejidad Computacional

Para estas pruebas se omitieron aquellas líneas de código que imprime el arreglo en cada iteración así como las mezclas particiones y creaciones del Heap -según sea el caso- de tal manera que solo muestra el arreglo original , el arreglo ordenado y el número de operaciones realizadas, esto con el fin de agilizar el proceso de ejecución principalmente en los arreglos de 1000,2000, 5000 y 10000 elementos.

Para contabilizar las operaciones se utilizó una variable llamada *contador* definida desde la main y pasando su dirección al menú de ordenamientos de esta manera mediante el uso de indirecciones podemos modificar su valor en cualquier punto de los algoritmos.

HeapSort

Con 50 elementos

```
El arreglo es: 486 847 209 588 464 870 867 842 824 908 647 832 325 244 883 527 983 778 304 627 751 37 704 555 330 84 308 378 495 179 969 99 961 270 332 633 590 549 948
325 497 848 31 635 765 269 359 519 891 888

-------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   4

---HeapSort---

El arreglo ordenado es: 31 37 84 99 179 209 244 269 270 304 308 325 325 330 332 359 378 464 486 495 497 519 527 549 555 588 590 627 633 635 647 704 751 765 778 824 832
842 847 848 867 870 883 888 891 908 948 961 969 983
Numero de elementos del arreglo: 50
El numero de operaciones es: 3151
Process returned 0 (0x0)   execution time : 3.658 s
Press any key to continue.
```

Num. Operaciones realizadas: 3151

Con 500 elementos

```
El arreglo es: 957 615 222 542 977 672 651 783 218 368 495 863 618 916 212 963 557 17 707 729 716 546 533 521 804 306 94 901 47 354 931 869 866 989 477 165 942 185 903
435 706 449 133 355 81 241 965 686 416 938 906 539 843 415 430 100 225 295 576 764 293 19 671 999 6 930 630 139 881 84 88 254 397 580 813 464 231 201 566 610 264 14 333
 647 964 857 97 282 636 609 369 616 670 628 270 471 738 87 462 536 465 775 710 385 514 744 384 837 85 868 294 480 379 686 861 630 165 728 317 907 172 641 792 992 297 31
5 905 877 639 924 632 738 733 635 299 667 568 742 102 131 793 514 222 39 331 514 234 678 903 923 131 280 651 904 38 741 592 257 544 807 731 97 506 443 255 544 940 332 9
13 494 862 616 572 682 445 423 366 410 603 761 163 75 371 864 568 724 475 486 570 598 178 479 488 310 788 658 502 350 676 76 492 40 969 806 471 117 852 697 718 174 529
391 599 785 356 819 939 621 806 163 79 217 288 573 597 690 359 936 474 940 409 524 89 308 599 61 136 603 587 620 530 725 211 926 861 922 735 465 692 165 216 557 799 36
681 159 729 951 507 140 307 404 227 459 544 577 870 51 49 660 49 880 465 104 258 968 499 561 643 819 507 578 992 458 242 637 654 7 238 923 391 678 433 332 315 362 377 2
49 9 458 520 8 430 93 101 935 917 501 131 813 473 824 208 206 295 204 288 181 527 348 227 559 347 502 885 821 345 872 20 74 414 578 695 406 716 36 997 730 309 979 204 7
57 944 585 675 975 764 747 346 260 411 39 944 699 353 413 293 703 12 93 193 693 126 177 549 815 376 949 697 889 266 476 765 978 265 668 164 470 608 46 726 969 442 167 9
25 766 254 218 354 777 804 289 529 866 592 177 658 3 677 230 889 395 955 71 928 260 899 36 452 858 210 373 269 678 245 212 618 73 845 573 364 117 699 653 642 704 133 73
7 198 402 673 977 317 890 489 461 636 195 639 110 225 965 131 544 184 953 236 582 687 53 788 375 452 297 231 536 829 673 476 330 641 716 140 909 333 60 851 581 969 807
229 476 692 661 35 456 509 403 390 150 12 720 619 903 576 326 424 754 907 891 526 507 827 688 183 514 577 786 434 215
```

```
-------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   4

---HeapSort---

El arreglo ordenado es: 3 6 7 8 9 12 12 14 17 19 20 35 36 36 36 38 39 39 40 46 47 49 49 51 53 60 61 71 73 74 75 76 79 81 84 85 87 88 89 93 93 94 97 97 100 101 102 104 1
10 117 117 126 131 131 131 131 133 133 136 139 140 140 150 159 163 163 164 165 165 165 167 172 174 177 177 178 181 183 184 185 193 195 198 201 204 204 206 208 210 211 2
12 212 215 216 217 218 218 222 222 225 225 227 227 229 230 231 231 234 236 238 241 242 245 249 254 254 255 257 258 260 260 264 265 266 269 270 280 282 288 288 289 293 2
93 294 295 295 297 297 299 306 307 308 309 310 315 315 317 317 326 330 331 332 332 333 333 345 346 347 348 350 353 354 354 355 356 359 362 364 366 368 369 371 373 375 3
76 377 379 384 385 390 391 391 395 397 402 403 404 406 409 410 411 413 414 415 416 423 424 430 430 433 434 435 442 443 445 449 452 452 456 458 458 459 461 462 464 465 4
65 465 470 471 471 473 474 475 476 476 476 477 479 480 486 488 489 492 494 495 499 501 502 502 506 507 507 507 509 514 514 514 514 520 521 524 526 527 529 529 530 533 5
36 536 539 542 544 544 544 544 544 546 549 557 557 559 561 566 568 568 570 572 573 573 576 576 577 577 578 578 580 581 582 585 587 592 592 597 598 599 599 603 603 608 609 6
10 615 616 616 618 618 619 620 621 628 630 630 632 635 636 636 637 639 639 641 641 642 643 647 651 651 653 654 658 658 660 661 667 668 670 671 672 673 673 675 676 677 6
78 678 678 681 682 686 686 687 688 690 692 692 693 695 697 697 699 699 703 704 706 707 710 716 716 716 718 720 724 725 726 728 729 729 730 731 733 735 737 738 738 741 7
42 744 747 754 757 761 764 764 765 766 775 777 783 785 786 788 788 792 793 799 804 804 806 806 807 807 813 813 815 819 819 821 824 827 829 837 843 845 851 852 857 858 8
61 861 862 863 864 866 866 868 869 870 872 877 880 881 885 889 889 890 891 899 901 903 903 903 904 905 906 907 907 909 913 916 917 922 923 923 924 925 926 928 930 931 9
35 936 938 939 940 940 942 944 944 949 951 953 955 957 963 964 965 965 968 969 969 969 975 977 977 978 979 989 992 992 997 999
Numero de elementos del arreglo: 500
El numero de operaciones es: 52538
```

Num. Operaciones realizadas: 52538

Con 1000 elementos

```
El arreglo es: 462 168 603 79 73 166 734 355 234 649 777 300 530 590 967 968 391 903 252 632 410 886 884 431 713 301 161 696 853 852 37 383 988 59 660 147 993 176 798 6
50 82 331 811 173 690 761 903 670 185 774 713 936 833 925 791 413 618 293 204 182 792 692 652 998 883 423 719 879 688 22 109 106 966 954 25 793 427 144 617 488 432 882
125 611 538 524 101 33 459 700 488 642 25 453 858 772 964 150 453 60 998 679 4 908 758 91 276 175 421 805 644 165 789 220 559 99 863 24 815 554 332 194 492 364 574 415
701 682 960 174 870 785 59 554 843 934 10 209 131 648 344 774 81 677 593 972 263 384 46 972 884 473 549 33 568 181 601 362 21 399 813 856 153 463 143 96 241 458 27 659
495 83 4 820 279 625 163 138 359 760 625 183 125 140 871 895 710 539 792 45 451 755 685 139 894 277 818 257 374 984 23 608 984 664 578 983 615 840 498 373 373 605 735 2
08 538 141 272 510 83 361 973 490 111 511 183 89 559 259 20 421 676 174 503 417 549 119 0 503 720 778 352 919 241 534 808 336 450 988 178 260 266 180 31 223 912 890 435
 888 17 681 518 779 297 854 348 494 86 257 343 891 787 896 388 902 772 548 179 591 677 583 763 882 704 310 222 741 559 508 944 377 318 664 142 725 253 312 189 903 133 4
29 512 164 496 504 22 803 182 654 482 37 365 64 349 999 488 374 562 275 475 446 736 676 846 407 557 153 281 226 873 60 631 763 473 538 854 149 349 989 630 404 180 633 3
75 727 543 589 648 541 507 563 949 174 514 865 465 350 564 464 140 742 466 500 863 236 564 501 822 764 451 414 612 198 569 685 366 258 96 746 552 826 737 75 685 863 966
 71 374 550 722 26 384 395 943 880 586 599 183 277 919 751 623 685 802 213 81 695 15 811 514 687 514 485 372 556 482 374 228 79 181 70 262 478 818 722 799 976 435 916 5
64 826 690 437 73 993 509 168 677 136 29 233 516 980 251 390 292 950 396 733 652 794 671 54 178 430 44 578 980 916 912 564 306 611 470 543 298 419 456 632 915 752 134 5
33 734 313 590 654 513 705 508 891 38 701 732 959 365 224 918 355 109 948 929 462 536 833 316 483 936 163 1 412 956 23 82 878 552 29 760 4 348 773 386 496 403 179 963 2
48 710 477 319 70 26 207 575 842 520 196 611 907 380 776 912 932 24 164 883 900 802 58 940 944 475 841 693 357 513 723 69 1 311 773 688 433 91 950 810 249 353 604 228 5
51 934 932 598 160 480 664 102 159 300 910 408 523 871 334 814 745 513 948 625 74 487 200 417 319 41 250 1 1 195 1 831 149 191 421 875 73 442 443 797 840 542 290 84 636
 417 948 156 557 836 823 688 868 462 349 993 61 898 600 157 501 38 66 339 599 885 156 198 494 704 229 816 894 240 718 601 419 751 579 152 919 592 204 882 94 756 13 624
389 831 83 461 458 10 681 233 957 738 169 432 559 49 950 31 130 91 833 159 857 651 627 562 353 111 531 774 545 25 24 80 494 760 434 202 362 767 532 712 741 684 844 312
413 932 270 569 818 469 933 901 245 956 763 97 610 168 900 572 19 222 182 382 960 54 436 844 780 565 739 305 633 877 220 751 814 339 892 350 160 967 360 612 423 105 861
 262 299 522 990 125 167 129 23 349 604 295 66 258 56 261 464 28 473 148 387 300 765 853 489 319 284 162 651 483 8 604 295 352 259 615 371 921 410 131 587 282 97 285 11
6 420 383 541 66 152 331 313 855 152 879 426 933 444 476 76 158 579 695 324 482 688 258 638 729 623 254 213 417 280 879 502 34 788 960 694 408 45 905 171 489 675 782 30
5 904 180 490 951 121 907 260 433 733 613 225 695 105 536 978 421 26 461 247 948 84 615 835 305 395 28 354 436 273 340 177 572 281 312 238 706 899 812 487 235 706 192 1
5 410 890 835 174 405 662 505 728 766 953 640 833 324 827 213 767 803 310 332 313 442 218 111 558 12 438 350 473 562 487 352 180 319 670 641 515 875 613 277 474 851 914
 932 321 537 848 503 391 184 290 402 749 428 65 874 220 991 104 680 161 382 670 974 432 174 539 178 216 145 599 505 637 317 359 510 515 791 33 827 402 707 323 247 589 8
72 215 688 455 210 850 371 504 727 580 730 716 711 786 77 501 893 379 636 226 955 800 260 640 69 464 600 208 949 338 738 249 719 260 666 352 221 979 398 765 694 52 83 8
31 114 51 915 978 561 682 679 713
```

```
-------------Menu-----------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   4

---HeapSort---

El arreglo ordenado es: 0 1 1 1 1 1 4 4 4 8 10 10 12 13 15 15 17 19 20 21 22 22 23 23 23 24 24 24 25 25 25 26 26 26 27 28 28 29 29 31 31 33 33 33 34 37 37 38 38 41 44 4
5 45 46 49 51 52 54 54 56 58 59 59 60 60 61 64 65 66 66 66 69 69 70 70 71 73 73 73 74 75 76 77 79 79 80 81 81 82 82 83 83 83 83 84 84 86 89 91 91 91 94 96 96 97 97 99 1
01 102 104 105 105 106 109 109 111 111 111 114 116 119 121 125 125 125 129 130 131 131 133 134 136 138 139 140 140 141 142 143 144 145 147 148 149 149 150 152 152 152 1
53 153 156 156 157 158 159 159 160 160 161 161 162 163 163 164 164 165 166 167 168 168 168 169 171 173 174 174 174 174 174 174 175 176 177 178 178 178 179 179 180 180 180 1
80 181 181 182 182 182 183 183 183 184 185 189 191 192 194 195 196 198 198 200 202 204 204 207 208 208 209 210 213 213 213 215 216 218 220 220 220 221 222 222 223 224 2
25 226 226 228 228 229 233 233 234 235 236 238 240 241 241 245 247 247 248 249 249 250 251 252 253 254 257 257 258 258 258 259 259 260 260 260 260 261 262 262 263 266 2
70 272 273 275 276 277 277 277 279 280 281 281 282 284 285 290 290 292 293 295 295 297 298 299 300 300 300 301 305 305 305 306 310 310 311 312 312 312 313 313 313 316 3
17 318 319 319 319 319 321 323 324 324 331 331 332 332 334 336 338 339 339 340 343 344 348 348 349 349 349 349 350 350 350 352 352 352 352 353 353 354 355 355 357 359 3
59 360 361 362 362 364 365 365 366 371 371 372 373 373 374 374 374 374 374 375 377 379 380 382 382 383 383 384 384 386 387 388 389 390 391 391 395 395 396 398 399 402 402 4
03 404 405 407 408 408 410 410 410 412 413 413 414 415 417 417 417 417 419 419 420 421 421 421 421 423 423 426 427 428 429 430 431 432 432 432 433 433 434 435 435 436 4
36 437 438 442 442 443 444 446 450 451 451 453 453 455 456 458 458 459 461 461 462 462 462 463 464 464 464 465 466 469 470 473 473 473 473 474 475 475 476 477 478 480 4
82 482 482 483 483 485 487 487 487 488 488 488 489 489 490 490 492 494 494 494 495 496 496 498 500 501 501 501 502 503 503 503 504 504 505 505 507 508 508 509 510 510 5
11 512 513 513 513 514 514 514 515 515 516 518 520 522 523 524 530 531 532 533 534 536 536 537 538 538 538 539 539 541 541 542 543 543 545 548 549 549 550 551 552 552 5
54 554 556 557 557 558 559 559 559 559 561 562 562 562 563 564 564 564 564 565 568 569 569 572 572 574 575 578 578 579 579 580 583 586 587 589 589 590 590 591 592 593 5
98 599 599 599 600 600 601 601 603 604 604 604 605 608 610 611 611 611 612 612 613 613 615 615 615 617 618 623 623 624 625 625 625 627 630 631 632 632 633 633 636 636 6
37 638 640 640 641 642 644 648 648 649 650 651 651 652 652 654 654 659 660 662 664 664 664 666 670 670 670 671 675 676 676 677 677 677 679 679 680 681 681 682 682 684 6
85 685 685 685 687 688 688 688 688 688 688 690 690 692 693 694 694 695 695 695 696 700 701 701 704 704 705 706 706 707 710 710 711 712 713 713 713 716 718 719 719 720 722 7
22 723 725 727 727 728 729 730 732 733 733 734 734 735 736 737 738 738 739 741 741 742 745 746 749 751 751 751 752 755 756 758 760 760 760 761 763 763 763 764 765 765 7
66 767 767 772 772 773 773 774 774 774 776 777 778 779 780 782 785 786 787 788 789 791 791 792 792 793 794 797 798 799 800 802 802 803 803 805 808 810 811 811 812 813 8
14 814 815 816 818 818 818 820 822 823 826 826 827 827 831 831 831 833 833 833 833 835 835 836 840 840 841 842 843 844 844 846 848 850 851 852 853 853 854 854 855 856 8
57 858 861 863 863 863 865 868 870 871 871 872 873 874 875 875 877 878 879 879 879 880 882 882 882 883 883 884 884 885 886 888 890 890 891 891 892 893 894 894 895 896 8
98 899 900 900 901 902 903 903 903 904 905 907 907 908 910 912 912 912 914 915 915 916 916 918 919 919 919 921 925 929 932 932 932 932 933 933 934 934 936 936 940 943 9
44 944 948 948 948 948 949 949 950 950 950 951 953 954 955 956 956 957 959 960 960 960 963 964 966 966 967 967 968 972 972 973 974 976 978 978 979 980 980 983 984 984 9
88 988 989 990 991 993 993 993 998 998 999
Numero de elementos del arreglo: 1000
El numero de operaciones es: 117712
```

Num. Operaciones realizadas: 117712

QuickSort

Con 500 elementos

```
El arreglo es: 239 659 971 449 779 735 635 350 181 693 275 682 441 217 839 202 125 964 762 414 351 617 33 181 422 789 15 661 684 465 292 995 449 401 427 792 363 450 612
 453 558 312 127 616 543 757 915 983 317 761 446 580 97 950 381 34 797 203 798 288 764 348 644 875 329 256 115 344 130 875 461 330 510 604 148 73 124 465 185 4 766 74 5
57 624 966 964 630 833 712 567 635 40 689 562 107 342 366 254 667 25 671 120 575 924 237 466 308 803 50 28 707 448 272 159 824 551 357 187 227 265 62 850 974 607 282 36
4 105 477 208 572 139 162 658 363 386 900 100 779 461 201 336 734 807 827 564 721 862 12 215 979 821 43 135 74 311 621 143 609 102 151 199 906 171 27 76 610 62 250 618
333 815 833 703 472 550 606 601 142 337 829 795 228 796 902 988 873 566 817 952 680 993 261 612 169 522 997 220 471 970 685 775 442 164 907 21 960 504 345 200 327 871 4
66 897 727 576 835 764 934 243 108 934 331 252 959 878 415 609 88 162 81 287 245 716 97 259 273 637 983 669 323 175 325 369 260 401 562 785 857 71 320 197 46 625 681 13
 188 125 757 128 259 287 15 622 460 743 506 363 416 508 157 388 613 736 702 927 259 964 422 550 777 982 617 709 590 871 146 685 917 178 684 503 807 290 843 620 879 395
92 933 3 604 235 903 79 19 708 689 78 553 596 737 427 192 88 875 804 298 742 179 601 484 203 438 740 411 268 364 849 597 729 9 567 740 495 492 840 854 37 455 278 448 66
6 480 126 861 981 911 233 609 548 145 792 491 233 292 169 738 58 509 968 207 504 751 823 740 587 172 40 235 139 912 594 474 586 116 523 309 992 689 989 53 247 197 228 2
72 680 547 382 832 581 374 628 776 489 206 121 45 757 463 80 617 373 565 331 534 906 21 985 321 265 432 127 189 884 225 739 593 199 33 308 404 269 320 710 785 668 7 917
 566 783 84 152 849 619 806 272 308 708 234 209 304 696 181 344 889 804 851 413 689 822 935 747 639 690 675 422 873 545 907 380 599 344 565 7 474 669 683 538 789 206 14
1 610 899 325 168 207 33 951 436 827 181 869 861 442 336 173 39 459 376 769 311 550 448 647 864 350 732 531 524 193
```

```
-------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   5

---QuickSort---

El arreglo ordenado es: 3 4 7 7 9 12 13 15 15 19 21 21 25 27 28 33 33 33 34 37 39 40 40 43 45 46 50 53 58 62 62 71 73 74 74 76 78 79 80 81 84 88 88 92 97 97 100 102 105
 107 108 115 116 120 121 124 125 125 126 127 127 128 130 135 139 139 141 142 143 145 146 148 151 152 157 159 162 162 164 168 169 169 171 172 173 175 178 179 181 181 181
 181 185 187 188 189 192 193 197 197 199 199 200 201 202 203 203 206 206 207 207 208 209 215 217 220 225 227 228 228 233 233 234 235 235 237 239 243 245 247 250 252 254
 256 259 259 259 260 261 265 265 268 269 272 272 272 273 275 278 282 287 287 288 290 292 292 298 304 308 308 308 309 311 311 312 317 320 320 321 323 325 325 327 329 330
 331 331 333 336 336 337 342 344 344 344 345 348 350 350 351 357 363 363 363 364 364 366 369 373 374 376 380 381 382 386 388 395 401 401 404 411 413 414 415 416 422 422
 422 427 427 432 436 438 441 442 442 446 448 448 448 449 449 450 453 455 459 460 461 461 463 465 465 466 466 471 472 474 474 477 480 484 489 491 492 495 503 504 504 506
 508 509 510 522 523 524 531 534 538 543 545 547 548 550 550 550 551 553 557 558 562 562 564 565 565 566 566 567 567 572 575 576 580 581 586 587 590 593 594 596 597 599
 601 601 604 604 606 607 609 609 609 610 610 612 612 613 616 617 617 617 618 619 620 621 622 624 625 628 630 635 635 637 639 644 647 658 659 661 666 667 668 669 669 671
 675 680 680 681 682 683 684 684 685 685 689 689 689 690 693 696 702 703 707 708 708 709 710 712 716 721 727 729 732 734 735 736 737 738 739 740 740 742 743 747
 751 757 757 757 761 762 764 764 766 769 775 776 777 779 779 783 785 785 789 789 792 792 795 796 797 798 803 804 804 806 807 807 815 817 821 822 823 824 827 827 829 832
 833 833 835 839 840 843 849 849 850 851 854 857 861 861 862 864 869 871 871 873 873 875 875 875 878 879 884 889 897 899 900 902 903 906 906 907 907 911 912 915 917 917
 924 927 933 934 934 935 950 951 952 959 960 964 964 964 966 968 970 971 974 979 981 982 983 983 985 988 989 992 993 995 997
Numero de elementos del arreglo: 500
El numero de operaciones es: 17142
```

Núm. Operaciones realizadas: 17142

Con 2000 elementos

```
El arreglo es: 457 625 169 334 697 124 734 378 737 948 806 243 445 605 996 783 272 62 276 530 289 891 800 398 220 652 706 388 844 317 690 484 56 183 459 250 116 911 769
 722 378 445 934 220 248 224 198 787 979 513 542 779 596 493 644 401 546 798 235 104 369 892 590 313 457 512 832 338 730 890 665 394 670 980 812 732 863 462 70 50 948 6
83 601 100 734 172 946 578 935 307 719 302 341 507 685 997 348 46 793 671 906 79 262 764 630 929 895 298 851 599 501 637 482 926 283 959 646 128 760 576 636 699 324 206
 746 705 695 960 807 360 502 264 860 629 672 760 433 188 296 151 315 748 115 636 741 577 541 153 517 161 951 67 288 172 803 449 576 165 379 956 916 246 471 6 817 987 43
8 566 952 334 175 90 672 83 504 622 885 188 447 428 145 600 350 516 459 679 564 607 901 97 466 686 378 16 564 961 80 667 913 389 986 389 884 628 881 149 300 499 467 223
 192 632 108 443 759 875 949 620 831 591 339 76 189 185 594 358 300 64 265 548 417 229 31 362 823 149 982 554 334 892 734 434 90 21 758 683 248 655 415 75 484 490 585 2
78 513 739 659 480 553 892 532 342 238 343 975 477 815 444 559 739 943 985 863 717 803 794 61 753 627 857 744 645 754 31 344 407 145 124 829 38 9 196 586 361 704 11 171
 639 38 177 973 742 967 985 189 952 884 255 763 75 206 577 752 840 994 887 167 839 524 213 288 136 317 369 674 213 342 536 130 502 482 958 431 437 214 589 512 3 237 117
 384 979 498 325 648 158 676 298 150 942 361 598 917 95 172 210 661 642 185 247 292 636 100 264 685 294 764 284 792 415 117 732 522 319 286 480 871 773 856 215 933 756
337 894 755 217 392 392 418 920 537 495 26 127 123 450 496 851 665 119 150 946 438 627 859 89 628 136 485 383 782 694 469 849 636 278 306 487 215 994 361 455 134 425 16
3 209 977 810 6 475 236 94 937 574 462 203 851 576 10 950 787 595 749 504 253 389 410 645 699 106 930 505 218 719 752 668 695 401 364 280 602 134 230 304 361 489 806 26
7 496 204 193 65 464 0 943 981 861 377 850 220 61 488 481 53 798 17 134 417 7 598 949 437 513 627 207 565 399 333 974 36 876 924 441 186 900 872 666 172 878 179 350 714
 376 35 164 941 220 251 66 352 713 986 212 728 981 356 695 255 882 728 53 597 614 58 986 504 408 2 111 998 271 229 36 632 861 57 163 578 844 391 994 944 592 660 985 547
 93 552 957 467 106 980 557 447 879 757 563 4 459 558 53 19 712 21 77 734 891 620 245 779 647 539 407 689 345 932 756 155 714 743 763 164 296 144 44 26 918 773 646 461
757 764 390 970 26 421 73 803 538 235 603 537 380 329 297 877 838 98 364 226 787 826 847 720 835 419 499 341 523 154 977 140 630 220 904 691 508 355 431 183 267 630 709
 483 875 991 623 513 166 902 770 72 800 63 450 375 571 295 258 113 262 317 733 329 871 814 732 735 70 353 761 940 4 981 227 122 741 359 83 936 818 536 286 181 785 247 8
4 858 744 171 241 933 552 508 898 875 663 810 130 990 401 774 405 695 608 680 405 902 897 96 643 968 208 177 880 376 87 538 229 385 291 663 792 802 352 576 655 883 422
529 474 287 732 809 212 99 860 567 378 606 343 466 472 225 682 215 568 961 361 929 564 267 843 371 70 359 398 666 694 712 957 897 28 331 375 252 302 286 938 2 868 963 8
7 797 778 348 48 955 558 577 807 474 114 768 961 343 892 885 507 520 437 747 771 667 103 233 955 726 226 952 249 721 249 340 986 481 968 1 737 222 641 272 938 366 31 48
2 828 75 658 836 574 251 581 699 529 200 972 746 102 919 578 818 130 212 482 282 439 814 724 103 413 750 962 269 809 326 43 486 628 280 656 539 967 268 254 465 357 374
955 29 299 119 684 92 539 947 182 900 623 852 460 367 900 29 553 433 788 441 78 423 705 652 526 498 207 346 391 37 777 751 28 883 567 446 841 773 670 79 15 787 561 555
561 760 950 993 798 499 324 630 567 651 564 336 563 322 821 324 130 894 5 641 751 284 476 389 849 421 875 974 603 406 53 179 683 43 244 28 736 380 104 647 448 858 404 9
25 880 331 113 876 877 308 951 38 571 311 673 730 780 318 432 217 851 862 211 812 450 231 622 965 224 395 432 843 623 221 585 457 477 180 4 273 241 808 689 149 57 221 8
14 86 138 766 87 841 87 131 692 143 525 901 154 155 775 201 157 634 501 940 932 494 242 702 421 619 870 714 450 630 129 677 404 210 234 397 440 860 759 668 999 690 5 58
1 538 96 483 934 901 35 762 774 660 506 879 137 604 618 612 602 467 352 595 216 490 380 813 332 857 972 363 498 876 342 491 662 701 803 332 833 371 896 749 972 449 183
283 364 641 270 354 838 59 941 383 13 664 246 659 946 564 624 354 820 916 748 621 720 196 436 428 546 812 896 152 805 140 428 632 135 861 711 418 49 546 932 237 957 831
 518 756 409 16 944 5 243 254 227 235 197 806 675 225 595 18 316 553 338 500 360 292 915 207 373 162 881 640 845 324 504 206 46 128 910 294 275 172 14 338 858 911 714 7
07 880 69 441 628 498 345 890 993 549 216 736 864 56 915 774 518 880 165 821 549 204 766 435 477 179 671 790 393 627 727 226 771 627 654 423 290 387 888 574 243 314 708
 109 862 624 277 426 979 758 400 180 165 970 631 688 269 431 366 99 258 45 268 707 799 477 29 283 733 759 277 758 571 119 727 115 779 5 47 554 824 341 253 263 205 687 7
23 468 839 487 240 644 245 991 622 621 750 668 897 405 228 991 114 833 224 180 51 314 271 32 299 905 475 246 603 923 61 995 474 296 745 323 399 537 282 947 309 993 817
657 548 814 592 521 501 984 843 697 646 833 1 295 839 12 54 306 684 659 973 902 124 716 14 59 369 228 74 920 657 871 582 117 156 710 719 390 743 650 67 783 761 744 713
292 212 670 609 799 846 69 262 712 880 831 827 568 959 642 172 822 193 22 23 934 18 164 51 703 184 64 392 764 836 296 695 813 698 693 650 687 562 465 719 865 920 233 66
 936 400 46 601 861 487 311 884 214 14 425 572 699 368 945 314 721 126 73 83 716 849 921 315 448 656 264 521 329 922 540 770 932 46 878 536 846 769 89 531 507 912 612 3
44 145 878 574 954 646 545 834 486 760 416 219 204 476 607 290 619 976 839 693 443 77 295 986 625 415 80 965 262 583 367 254 209 179 500 718 895 52 819 980 896 464 192
707 843 913 713 542 741 974 54 823 214 33 276 859 364 273 195 338 300 656 621 518 86 768 520 475 881 825 101 854 311 254 169 832 922 585 156 125 231 556 529 919 210 754
 629 857 562 492 860 401 102 696 800 749 904 712 731 658 817 911 127 369 813 849 843 763 592 763 9 550 674 88 886 160 415 102 418 515 34 977 553 834 211 814 385 395 974
 642 661 877 606 628 752 916 25 569 214 459 630 239 928 42 269 4 23 584 46 927 827 778 212 462 795 108 707 642 976 615 848 792 573 764 8 586 722 992 116 30 18 905 811 9
49 225 624 456 798 106 951 478 523 391 693 537 517 306 488 759 492 60 138 390 457 469 742 248 536 363 965 565 520 726 108 535 349 27 33 292 336 735 278 783 404 999 21 5
28 772 542 725 997 660 479 403 545 302 619 956 245 982 892 914 932 87 315 101 667 39 120 877 608 544 784 909 425 297 427 857 249 997 230 596 520 122 423 491 800 285 264
 646 543 107 645 792 553 379 497 404 461 907 638 939 12 720 395 907 333 610 571 242 309 255 836 204 45 606 741 585 188 503 380 771 918 849 310 190 309 644 720 632 639 7
53 606 875 729 455 914 665 363 271 679 88 395 209 558 606 758 752 97 582 820 812 214 83 307 823 231 341 159 731 935 890 617 825 631 750 989 941 187 62 297 352 118 714 7
39 125 333 417 2 953 811 713 816 509 216 359 341 629 31 884 375 550 64 1 666 124 211 36 107 969 799 3 313 445 146 848 117 794 833 278 585 47 554 924 144 112 104 399 277
 548 63 636 774 847 65 422 826 96 960 309 936 165 605 416 818 822 369 92 618 680 770 853 172 323 119 492 85 814 103 258 138 74 727 517 179 964 99 24 993 638 848 623 878
 216 663 252 148 203 748 219 641 775 519 61 962 54 418 411 158 872 637 785 804 688 543 622 566 79 830 776 296 863 961 605 412 178 299 23 264 61 362 880 363 671 265 615
```

```
---QuickSort---

El arreglo ordenado es: 0 1 1 1 2 2 2 3 3 4 4 4 4 5 5 5 5 6 6 7 8 9 9 10 11 12 12 12 13 14 14 14 15 16 16 17 18 18 18 19 21 21 21 21 22 23 23 23 24 24 25 26 26 26 27 27
 28 28 28 29 29 29 29 30 31 31 31 31 32 33 33 34 35 35 35 36 36 36 37 38 38 38 39 42 43 43 44 45 45 45 46 46 46 46 46 47 47 48 49 50 51 51 52 53 53 53 53 54 54 54 56 56
 57 57 58 59 59 60 61 61 61 61 61 62 62 63 63 64 64 64 65 65 66 66 67 67 69 69 70 70 70 72 73 73 74 74 75 75 75 76 77 77 78 79 79 79 80 80 83 83 83 83 84 85 86 86 86 87
 87 87 87 87 88 88 88 89 89 90 90 92 92 92 92 93 94 95 96 96 96 97 97 97 98 99 99 99 100 100 101 101 102 102 102 102 103 103 103 103 104 104 104 106 106 106 107 107 108
 108 108 109 111 112 113 113 114 114 115 115 116 116 117 117 117 117 118 119 119 119 119 120 120 122 122 123 123 124 124 124 124 125 125 126 127 127 128 128 129 130 130
 130 130 131 134 134 134 134 135 136 136 137 138 138 138 140 140 141 143 144 144 145 145 145 146 148 149 149 149 150 150 151 152 153 154 154 155 155 156 156 157 158 158
 159 160 160 161 162 163 163 164 164 164 165 165 165 165 166 167 169 169 170 171 171 172 172 172 172 172 172 172 175 175 177 177 178 178 179 179 179 179 179 180 180 180
 181 182 183 183 183 184 185 185 186 187 188 188 188 189 189 190 190 192 192 193 193 194 195 196 196 197 198 198 200 201 203 203 204 204 204 204 205 206 206 206 207 207
 207 208 209 209 209 210 210 210 211 211 211 211 212 212 212 212 212 213 213 214 214 214 214 214 215 215 215 216 216 216 216 217 217 218 219 219 220 220 220 220 220 221
 221 221 221 221 222 223 224 224 224 225 225 225 225 226 226 226 227 227 228 228 229 229 229 230 230 231 231 231 232 233 233 234 235 235 235 236 237 237 238 239 240 241
 241 241 242 242 243 243 243 243 244 245 245 245 246 246 246 247 247 248 248 248 249 249 249 250 251 251 252 252 253 253 254 254 254 254 254 255 255 255 258 258 258 262
 262 262 262 263 264 264 264 264 264 264 265 265 267 267 267 268 268 269 269 269 270 271 271 271 272 272 273 273 275 275 276 276 277 277 277 278 278 278 278 280 280 282
 282 283 283 283 284 284 285 286 286 286 287 288 288 288 289 290 290 291 292 292 292 292 294 294 295 295 295 296 296 296 296 296 297 297 297 298 298 299 299 299 300 300 300
 300 302 302 302 303 304 306 306 306 307 307 308 309 309 309 309 309 310 311 311 311 313 313 314 314 314 315 315 315 316 317 317 317 318 319 321 322 322 323 323 324 324
 324 324 325 325 326 329 329 329 331 331 332 332 333 333 333 334 334 334 336 336 337 338 338 338 338 339 340 341 341 341 341 341 342 342 342 343 343 343 344 344 345 345
 345 346 348 348 349 350 350 352 352 352 352 353 353 354 354 355 356 357 358 359 359 359 360 360 361 361 361 361 362 362 363 363 363 363 364 364 364 364 366 366 367
 367 368 369 369 369 369 369 371 371 373 374 375 375 375 376 376 377 378 378 378 378 379 379 380 380 380 380 382 383 383 384 385 385 387 388 389 389 389 389 390 390 390
 391 391 391 392 392 392 393 394 395 395 395 395 395 396 397 398 398 399 399 399 400 400 401 401 401 401 403 404 404 404 404 404 405 405 405 406 407 407 408 409
 410 411 412 413 415 415 415 415 416 416 417 417 417 418 418 418 418 418 419 421 421 421 422 422 423 423 423 423 425 425 425 426 427 428 428 428 430 431 431 431 432 432
 433 433 434 435 436 437 437 437 437 438 438 439 440 441 441 441 443 443 444 445 445 445 445 445 446 447 447 448 448 449 449 450 450 450 450 455 455 455 456 457 457 457 457
 459 459 459 459 460 461 461 462 462 462 464 464 465 465 466 466 466 467 467 467 468 469 469 471 472 474 474 474 475 475 475 476 476 477 477 477 477 478 479 479 480 480
 481 481 482 482 482 482 482 483 483 483 484 484 485 486 486 486 487 487 487 488 488 489 490 490 491 491 492 492 492 493 494 495 496 496 497 498 498 498 498 499 499 499
 499 500 500 501 501 501 502 502 502 503 504 504 504 504 505 506 507 507 507 508 508 509 512 512 513 513 513 513 515 516 517 517 517 518 518 518 519 520 520 520 520 521
 521 522 523 523 524 525 526 528 529 529 529 530 531 532 535 536 536 536 536 537 537 537 537 538 538 538 539 539 539 540 541 542 542 542 543 543 544 545 545 546 546 546
 546 547 548 548 548 549 549 550 550 552 552 553 553 553 553 553 553 554 554 554 555 556 557 558 558 558 559 561 561 562 562 563 563 564 564 564 564 564 565 565 566 566 567
 567 567 568 568 569 571 571 571 571 572 573 574 574 574 574 575 576 576 576 576 577 577 577 578 578 578 578 581 581 582 582 583 584 585 585 585 585 585 586 586 589 589
 590 591 592 592 592 593 594 595 595 595 596 596 597 597 598 598 599 600 601 601 602 602 603 603 603 604 605 605 605 606 606 606 606 606 607 607 608 608 609 610 612 612
 612 614 615 615 616 617 618 618 619 619 619 619 619 620 620 621 621 621 622 622 622 622 623 623 623 623 624 624 624 624 625 625 625 627 627 627 627 627 628 628 628 628
 628 629 629 629 630 630 630 630 630 630 631 631 632 632 632 632 634 636 636 636 636 636 636 637 637 638 638 639 639 640 641 641 641 641 642 642 642 642 642 643 644 644 644
 645 645 645 645 646 646 646 646 646 647 647 648 650 650 651 652 652 654 655 655 656 656 656 657 657 658 658 658 659 659 659 660 660 660 661 661 661 662 663 663 663 664
 665 665 665 665 666 666 666 667 667 667 668 668 668 669 670 670 670 671 671 671 672 672 673 674 674 675 676 677 679 679 680 680 681 682 683 683 683 684 684 685 685 686
 687 687 688 688 689 689 689 690 690 691 692 693 693 693 694 694 695 695 695 695 695 695 696 697 697 698 699 699 699 699 701 702 703 704 705 705 706 707 707 707 707 708
 708 709 710 711 712 712 712 712 713 713 713 713 714 714 714 714 714 714 716 716 717 718 719 719 719 719 720 720 720 720 721 721 722 722 723 724 725 726 726 727 727 727 728
 728 729 730 730 731 731 732 732 732 732 733 733 734 734 734 734 735 735 736 736 737 737 739 739 739 741 741 741 741 742 742 743 743 744 744 744 745 746 746 747 748 748
 748 749 749 749 750 750 750 751 751 752 752 752 752 753 753 754 754 755 756 756 756 757 757 758 758 758 758 759 759 759 759 760 760 760 760 761 761 762 763 763 763 763
 764 764 764 764 764 766 766 768 768 768 769 769 769 770 770 770 771 771 771 772 773 773 773 774 774 774 774 775 775 776 777 778 778 779 779 779 780 782 783 783 783 784
 785 785 785 787 787 787 787 788 790 792 792 792 792 793 794 794 795 795 797 798 798 798 798 799 799 799 800 800 800 800 802 803 803 803 803 804 805 805 806 806 806 807 807
 808 809 809 810 810 811 811 811 812 812 812 812 813 813 813 814 814 814 814 814 814 814 815 816 817 817 817 818 818 818 819 820 820 821 821 821 822 822 823 823 823 824 825
 825 826 826 827 827 827 828 829 830 831 831 831 832 832 832 833 833 833 833 834 834 835 836 836 836 837 838 838 839 839 839 839 840 841 841 843 843 843 843 843 844 844
 845 846 846 847 847 848 848 848 849 849 849 849 849 850 851 851 851 851 851 852 853 854 856 857 857 857 857 858 858 858 859 859 860 860 860 860 861 861 861 861 862 862
```

```
 863 863 863 864 865 868 870 871 871 871 872 872 875 875 875 875 875 876 876 876 877 877 877 877 877 878 878 878 878 879 879 880 880 880 880 880 880 881 881 881 882 883
 883 884 884 884 884 885 885 886 886 887 888 890 890 890 891 891 892 892 892 892 892 894 894 895 895 896 896 896 896 897 897 897 898 900 900 900 901 901 901 902 902 902
 904 904 905 905 906 907 907 909 910 911 911 911 912 913 913 914 914 915 915 916 916 916 916 917 918 918 919 919 919 920 920 920 921 922 922 923 924 924 925 926 927 928
 929 929 930 932 932 932 932 932 933 933 934 934 934 935 935 936 936 936 937 938 938 939 940 940 941 941 941 942 943 943 944 944 945 946 946 946 947 947 948 948 949 949
 949 949 950 950 951 951 951 952 952 952 953 954 955 955 955 955 955 955 956 956 957 957 957 958 959 959 960 960 961 961 961 961 962 962 963 964 965 965 965 967 967 968 968
 969 970 970 972 972 972 973 973 974 974 974 974 975 976 976 977 977 977 979 979 979 980 980 980 981 981 981 982 982 984 984 985 985 985 986 986 986 986 986 987 989 990
 991 991 991 992 993 993 993 993 994 994 994 995 996 997 997 997 997 998 999 999
Numero de elementos del arreglo: 2000
El numero de operaciones es: 91175
Process returned 0 (0x0)   execution time : 2.744 s
Press any key to continue.
```

Núm. Operaciones realizadas 91175

MergeSort

Con 100 elementos

```
El arreglo es: 498 296 570 961 245 193 993 868 728 48 825 66 369 855 413 559 274 535 94 182 312 591 257 759 887 655 37 39 301 172 187 856 756 572 454 419 276 157 374 49
1 561 912 480 60 704 860 736 476 15 825 618 824 318 425 260 122 209 366 664 954 655 72 737 587 872 757 332 272 994 502 202 230 980 192 738 91 79 272 628 847 842 51 611
378 485 621 331 151 961 63 740 87 534 335 84 33 84 859 140 976

--------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   6

---MergeSort---

El arreglo ordenado es: 15 33 37 39 48 51 60 63 66 72 79 84 84 87 91 94 122 140 151 157 172 182 187 192 193 202 209 230 245 257 260 272 272 274 276 296 301 312 318 331
332 335 366 369 374 378 413 419 425 454 476 480 485 491 498 502 534 535 559 561 570 572 587 591 611 618 621 628 655 655 664 704 728 736 737 738 740 756 757 759 824 825
825 842 847 855 856 859 860 868 872 887 912 954 961 961 976 980 993 994
Numero de elementos del arreglo: 100
El numero de operaciones es: 4694
```

Num. Operaciones realizadas: 4694

Con 800 elementos

```
El arreglo es: 491 12 790 177 147 271 674 713 979 811 398 688 147 435 622 691 480 475 576 258 218 973 12 276 743 188 539 143 737 80 738 898 718 396 679 14 632 708 371 9
65 588 875 552 593 990 803 580 520 573 607 510 862 690 422 281 755 535 896 112 431 670 390 161 92 295 813 173 851 974 147 14 809 36 429 905 955 293 992 622 159 739 687
630 437 361 29 357 423 314 131 872 692 450 227 481 675 533 430 496 207 735 260 868 269 444 538 690 734 441 969 573 97 634 636 671 504 933 409 743 422 999 324 716 895 75
8 49 73 177 735 385 129 66 457 910 909 450 298 234 699 514 568 285 897 133 282 342 858 510 24 836 352 746 112 877 796 7 187 806 722 669 111 84 344 804 902 547 984 695 3
83 589 747 576 491 149 697 147 771 964 743 527 1 932 968 162 236 228 823 157 775 195 387 72 164 672 668 262 509 433 581 197 851 874 371 895 779 121 253 796 558 864 597
628 679 442 550 287 787 786 229 399 316 104 365 545 766 493 422 440 702 561 366 458 383 867 203 375 911 881 183 815 161 904 20 507 324 998 698 976 781 695 776 951 48 54
1 663 864 853 931 181 32 120 667 640 528 561 614 593 313 350 461 812 717 166 453 101 95 415 48 653 701 65 602 941 492 893 329 394 582 425 337 546 128 853 215 582 231 26
7 364 688 236 546 12 343 324 291 151 952 709 184 232 63 942 322 893 535 863 522 182 78 25 618 337 676 815 436 716 310 237 747 101 422 832 858 343 381 542 554 302 708 59
9 239 469 983 28 597 686 83 599 823 234 642 140 632 985 452 38 157 804 280 996 779 616 220 949 397 846 641 818 337 250 298 708 244 873 342 294 467 606 444 628 827 134 4
23 876 731 914 546 971 985 765 429 36 41 367 297 132 725 265 313 609 322 610 673 909 7 370 475 693 876 13 192 343 856 489 396 35 547 420 421 417 334 472 261 659 743 848
 894 620 849 0 925 754 788 431 274 768 553 544 127 647 600 770 463 831 909 493 410 656 545 838 566 173 605 685 991 483 996 292 274 46 761 34 182 101 383 33 333 479 424
333 450 967 784 711 419 804 666 920 488 395 251 474 956 665 182 747 172 203 904 667 726 737 133 240 892 43 50 240 873 234 541 24 467 466 330 938 473 438 95 972 295 917
794 134 360 780 362 427 67 292 11 909 561 485 390 611 342 51 103 76 942 848 113 24 610 42 815 524 299 838 541 431 671 501 135 131 105 798 423 542 813 533 853 993 348 70
6 634 720 389 909 917 372 310 601 490 398 721 366 344 45 897 803 98 201 791 899 832 977 472 464 95 330 186 22 870 96 447 281 186 34 832 409 933 625 838 933 546 258 277
190 620 813 987 525 847 308 311 248 279 991 69 449 856 616 601 399 798 435 241 249 582 672 112 960 493 139 113 212 63 908 401 810 178 643 848 893 780 662 456 923 931 15
9 759 585 100 111 650 617 741 656 371 114 807 801 309 977 702 418 276 271 579 235 387 835 924 321 16 722 642 993 394 916 843 503 430 633 201 649 798 939 868 447 36 773
595 246 910 957 800 238 962 334 309 713 403 417 57 901 52 292 855 683 701 460 673 320 764 816 753 644 453 173 409 698 925 49 596 719 43 170 859 111 361 319 684 236 258
18 145 408 540 508 61 237 55 401 372 570 158 518 27 401 358 672 501 103 749 998 440 980 478 506 16 748 385 308 890 53 703 416 284 369 87 737 883 278 83 366 796 253 343
515 47 344 202 99 41 23 13 533 209 981 924 325 5 438 811 844 254 227 596 162 242 50 326 275 392 16 279 752
```

```
--------------Menu------------
Elija el algortimo de ordenamiento
1)InsertionSort
2)SelectionSort
3)BubbleSort
4)HeapSort
5)QuickSort
6)MergeSort   6

---MergeSort---

El arreglo ordenado es: 0 1 5 7 7 11 12 12 12 13 13 14 14 16 16 16 18 20 22 23 24 24 24 25 27 28 29 32 33 34 34 35 36 36 36 38 41 41 42 43 43 45 46 47 48 48 49 49 50 50
 51 52 53 55 57 61 63 63 65 66 67 69 72 73 76 78 80 83 83 84 87 92 95 95 95 96 97 98 99 100 101 101 101 103 103 104 105 111 111 111 112 112 112 113 113 114 120 121 127
128 129 131 131 132 133 133 134 134 135 139 140 143 145 147 147 147 147 149 151 157 157 158 159 159 161 161 162 162 164 166 170 172 173 173 173 177 177 178 181 182 182
182 183 184 186 186 187 188 190 192 195 197 201 201 202 203 203 207 209 212 215 218 220 227 227 228 229 231 232 234 234 234 235 236 236 236 237 237 238 239 240 240 241
242 244 246 248 249 250 251 253 253 254 258 258 258 260 261 262 265 267 269 271 271 274 274 275 276 276 277 278 279 279 280 281 281 282 284 285 287 291 292 292 292 293
294 295 295 297 298 298 299 302 308 308 309 309 310 310 311 313 313 314 316 319 320 321 322 322 324 324 324 325 326 329 330 330 333 333 334 334 337 337 337 342 342 342
343 343 343 343 344 344 344 348 350 352 357 358 360 361 361 362 364 365 366 366 366 367 369 370 371 371 371 372 372 375 381 383 383 383 385 385 387 387 389 390 390 392
394 394 395 396 396 397 398 398 399 399 401 401 401 403 408 409 409 409 410 415 416 417 417 418 419 420 421 422 422 422 422 423 423 423 424 425 427 429 429 430 430 431
431 431 433 435 435 436 437 438 438 440 440 441 442 444 444 447 447 449 450 450 450 452 453 453 456 457 458 460 461 463 464 466 467 467 469 472 472 473 474 475 475 478
479 480 481 483 485 488 489 490 491 491 492 493 493 493 496 501 501 503 504 506 507 508 509 510 510 514 515 518 520 522 524 525 527 528 533 533 533 535 535 538 539 540
541 541 541 542 542 544 545 545 545 546 546 546 546 547 547 550 552 553 554 558 561 561 561 566 568 570 573 573 576 576 579 580 581 582 582 582 585 588 589 593 593 595 596
596 597 597 599 599 600 601 601 602 605 606 607 609 610 610 611 614 616 616 617 618 620 620 622 622 625 628 628 630 632 632 633 634 634 636 640 641 642 642 643 644 647
649 650 653 656 656 659 662 663 665 666 667 667 668 669 670 671 671 672 672 672 673 673 674 675 676 679 679 683 684 685 686 687 688 688 690 690 691 692 693 695 695 697
698 698 699 701 701 702 702 703 706 708 708 708 709 711 713 713 716 716 717 718 719 720 721 722 722 725 726 731 734 735 735 737 737 737 738 739 741 743 743 743 743 746
747 747 747 748 749 752 753 754 755 758 759 761 764 765 766 768 770 771 773 775 776 779 779 780 780 781 784 786 787 788 790 791 794 796 796 796 798 798 798 800 801 803
803 804 804 806 807 809 810 811 811 812 813 813 813 815 815 816 818 823 823 827 831 832 832 832 835 836 838 838 838 843 844 846 847 848 848 848 849 851 851 853
853 853 855 856 856 858 858 859 862 863 864 864 867 868 868 870 872 873 873 874 875 876 876 877 881 883 890 892 893 893 893 894 895 895 896 897 897 898 899 901 902 904
904 905 908 909 909 909 909 909 910 910 911 914 916 917 917 920 923 924 924 925 925 931 931 932 933 933 933 938 939 941 942 942 949 951 952 955 956 957 960 962 964 965
967 968 969 971 972 973 974 976 977 977 979 980 981 983 984 985 985 987 990 991 991 992 993 993 996 996 998 998 999
Numero de elementos del arreglo: 800
El numero de operaciones es: 51988
```

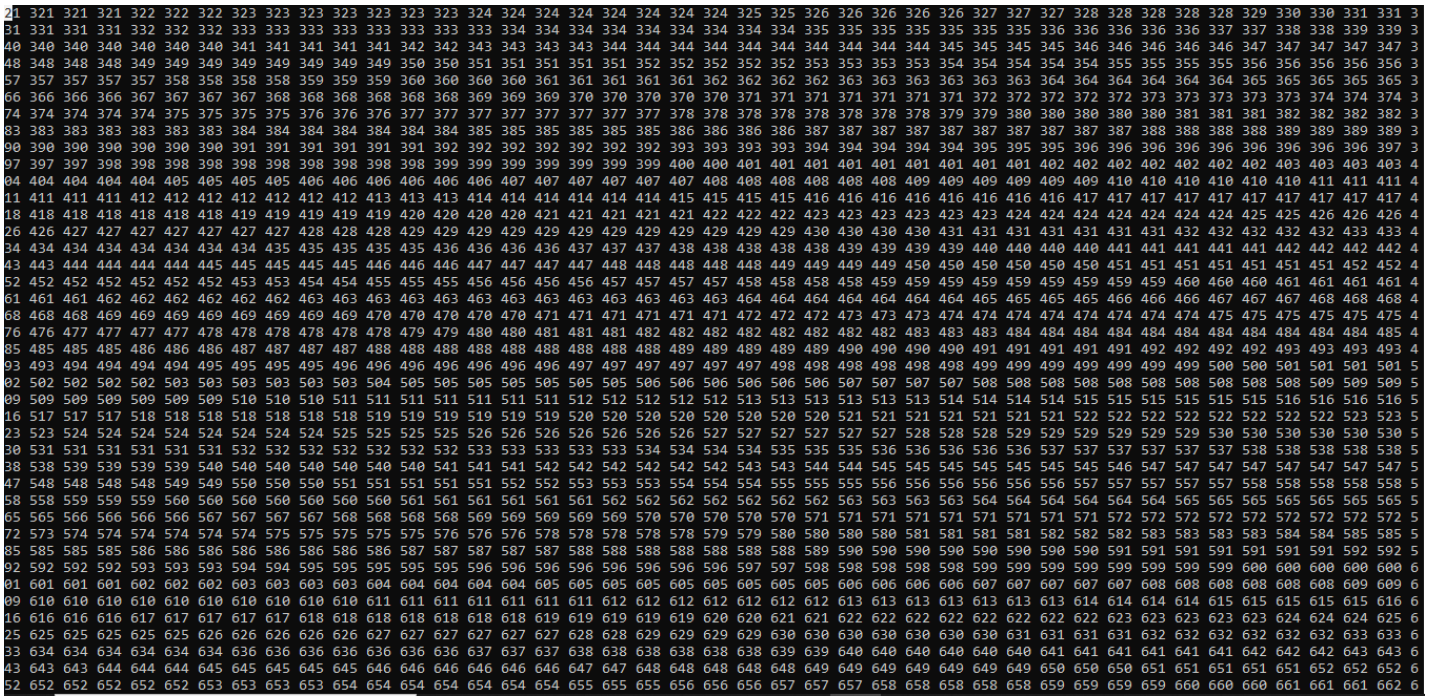
Num. Operaciones realizadas: 51988

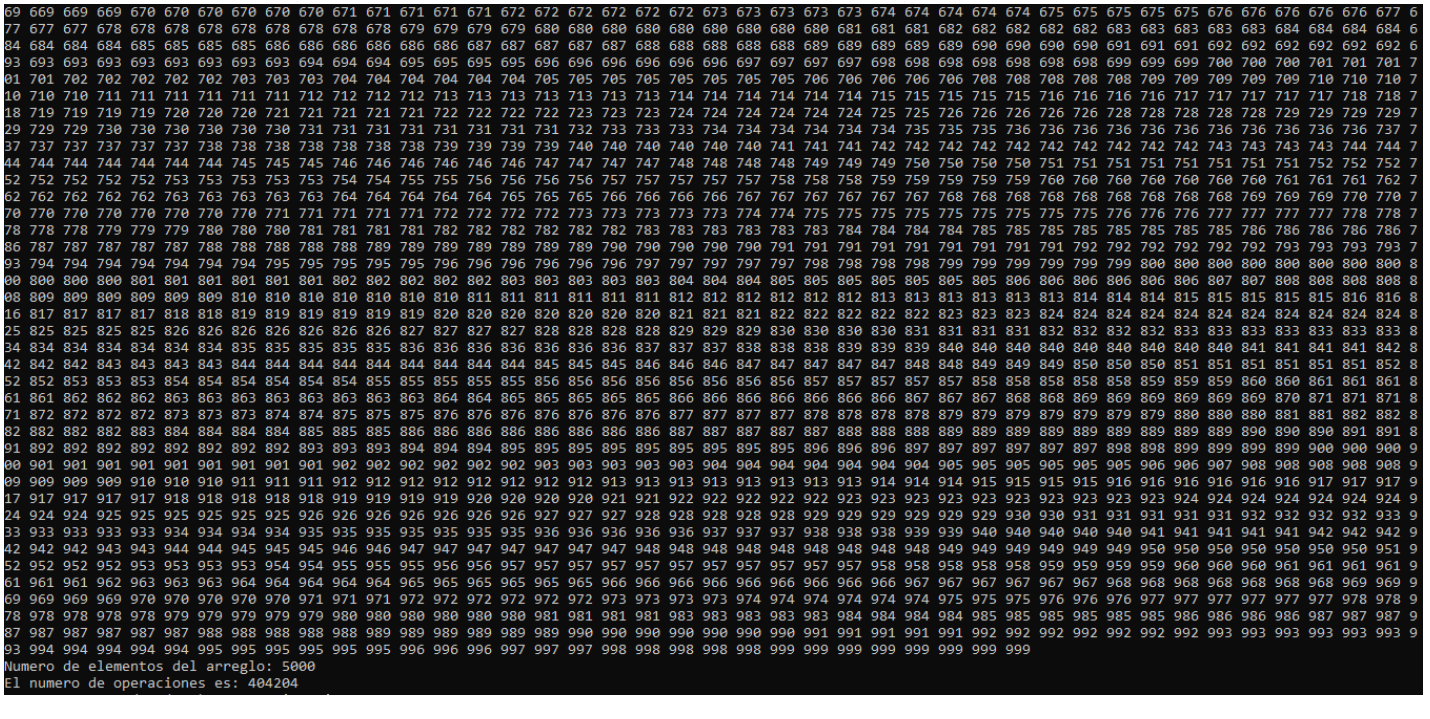
Con 5000 elementos

```
El arreglo es: 800 959 80 347 55 189 952 514 266 503 132 61 228 511 377 489 129 212 175 56 52 484 663 860 126 194 403 44 642 401 486 368 791 562 469 927 254 762 470 316
 76 968 664 948 146 461 612 531 835 613 267 776 766 541 328 278 477 582 231 12 877 113 693 799 31 739 802 78 398 994 453 190 530 720 934 375 724 592 522 439 499 327 258
 526 990 966 885 418 531 941 224 564 120 361 987 418 736 801 566 366 744 77 355 263 738 152 798 734 209 584 258 108 319 483 333 482 388 349 911 441 768 559 87 84 410 14
6 559 770 857 388 641 808 668 306 178 726 85 911 502 865 739 524 820 34 232 161 603 36 280 226 61 571 171 445 915 636 445 339 682 303 316 326 68 636 665 386 233 373 619
 134 528 360 800 837 217 313 678 0 109 261 105 238 304 633 693 631 269 268 503 525 634 762 71 300 886 531 585 428 377 291 196 749 515 491 1 426 304 873 832 664 49 211 8
88 969 150 49 558 940 99 12 521 856 81 489 515 721 991 923 162 721 9 472 639 243 31 978 302 532 713 938 376 75 553 392 310 106 889 309 190 989 592 812 787 427 159 894 3
12 19 512 158 253 889 841 668 277 824 805 526 319 338 156 636 188 710 981 290 864 282 657 499 285 113 89 364 940 742 618 752 397 418 736 330 530 499 897 767 406 493 260
 53 251 901 890 623 957 4 315 307 473 464 364 773 43 977 328 45 742 411 452 902 140 399 111 449 856 991 282 258 594 42 285 928 652 916 946 923 803 18 469 361 872 565 4
418 863 501 508 213 71 751 36 18 715 2 782 273 145 750 571 387 377 518 622 736 496 256 972 392 886 467 587 286 220 672 344 482 186 92 525 678 464 421 987 371 963 986 35
6 12 499 496 493 843 27 226 17 507 344 665 857 183 565 461 435 897 494 442 233 685 652 719 668 462 737 19 0 833 300 672 65 89 303 892 782 539 855 798 374 259 340 62 336
 118 474 760 247 226 326 243 294 506 173 141 688 39 434 249 83 340 574 403 606 144 901 243 952 825 374 585 219 687 51 240 922 14 211 757 182 521 164 314 999 795 396 606
 961 631 610 760 789 892 137 12 834 729 917 725 131 583 600 731 850 470 399 166 205 592 649 796 793 710 880 304 596 999 939 620 632 376 514 304 212 625 612 659 301 452
398 688 221 523 517 405 186 274 654 913 91 99 627 780 863 705 929 268 878 399 688 13 931 625 211 344 531 971 592 223 719 529 745 895 672 959 149 799 844 906 441 264 491
 686 259 870 336 634 656 135 948 569 268 105 93 250 186 680 463 730 637 912 265 696 711 976 12 569 719 896 367 694 808 340 166 35 87 18 548 305 989 8 991 990 39 9 739 7
31 978 522 274 766 347 887 506 905 934 204 775 563 724 401 227 414 851 205 651 193 465 662 836 46 512 864 966 902 97 325 423 372 314 19 307 775 57 404 475 788 174 794 1
15 748 396 345 824 290 857 794 562 271 943 805 159 561 393 86 913 682 809 980 392 853 969 430 914 54 537 206 826 962 545 67 191 68 167 249 487 994 163 684 60 751 345 46
9 173 843 687 468 880 536 744 622 591 606 608 649 64 497 129 294 985 489 277 750 244 227 45 193 471 950 993 394 616 408 233 167 231 562 572 207 532 505 974 461 586 743
551 516 732 298 378 995 464 882 723 748 289 601 701 744 497 262 227 317 478 618 365 202 76 839 574 377 308 926 611 958 412 128 702 744 206 247 564 358 5 406 587 836 958
 558 81 262 908 18 561 948 742 125 543 670 684 547 110 690 424 806 431 604 562 74 85 685 855 506 703 223 282 4 229 199 123 370 863 132 280 219 579 264 513 202 69 157 42
4 469 411 801 945 912 249 677 390 571 75 98 822 530 850 453 127 67 825 893 737 605 0 994 813 151 752 344 742 255 680 805 591 372 903 239 709 614 387 945 247 364 937 152
 563 374 262 705 66 509 974 917 295 967 412 279 940 248 369 909 158 757 383 14 177 474 824 404 407 647 363 737 877 878 56 654 615 928 931 122 557 617 94 144 605 457 986
 676 484 924 374 855 799 12 228 209 103 742 303 639 964 43 970 405 185 519 933 53 520 447 889 641 215 966 487 853 556 137 651 433 733 813 831 949 787 693 925 69 216 824
 52 928 926 969 736 20 306 145 659 972 673 715 600 989 353 46 346 197 811 977 536 284 368 761 155 368 482 998 459 836 524 815 207 128 844 355 924 983 768 59 977 372 7 6
52 800 383 978 96 678 271 578 471 39 224 300 66 884 695 626 265 86 3 267 29 488 271 351 800 304 949 795 316 214 693 468 242 728 62 490 268 917 276 378 717 296 339 297 4
71 904 155 205 892 30 324 542 761 966 260 622 433 610 599 540 654 355 478 383 301 361 595 753 466 77 356 578 177 181 670 127 472 760 379 819 113 483 343 676 524 603 176
 913 980 851 636 315 256 968 912 89 245 643 270 792 274 295 990 554 613 532 48 183 244 580 640 538 527 908 628 926 740 477 970 488 770 771 789 894 58 509 354 232 554 22
 398 558 261 827 370 227 616 999 866 88 181 380 850 588 29 274 770 475 586 788 771 460 195 313 810 215 676 785 805 238 77 986 84 700 209 817 794 364 146 660 235 859 890
 759 328 664 885 749 617 131 686 322 579 484 248 999 474 373 942 726 136 147 724 918 455 792 815 225 731 509 994 868 828 207 427 974 35 311 654 773 810 470 881 518 569
992 751 11 46 697 904 146 396 773 240 547 664 572 63 434 64 519 18 942 572 4 833 633 106 616 264 290 781 693 225 935 789 497 361 806 653 137 826 678 41 942 970 497 886
704 82 645 7 560 441 476 420 586 826 696 550 979 475 443 681 927 223 950 465 590 335 344 840 879 652 378 398 630 321 736 786 83 750 482 488 347 15 28 116 450 879 779 14
 344 879 417 223 764 596 363 922 103 992 321 478 731 847 653 203 515 836 128 355 460 629 153 76 901 629 663 891 251 144 82 323 968 668 596 374 957 511 558 203 717 19 40
0 0 964 785 265 834 895 357 576 307 590 30 218 211 792 769 413 306 341 222 156 940 650 331 636 663 417 887 806 681 191 170 751 22 778 59 324 608 610 787 258 447 78 408
124 508 667 452 118 640 537 463 978 970 271 490 929 105 288 790 965 911 525 260 812 255 301 478 599 416 810 165 12 537 766 509 440 688 182 830 281 966 46 669 340 925 71
7 66 834 741 571 571 313 619 800 241 224 899 822 832 978 856 657 829 255 129 422 988 87 182 842 352 448 147 4 318 922 188 494 348 753 403 32 493 179 869 942 592 899 947
 860 595 165 168 926 250 66 566 972 485 638 527 462 600 129 768 476 876 523 88 114 918 516 178 530 38 315 670 122 467 484 415 791 222 287 233 170 428 335 217 274 88 720
 666 98 456 341 277 221 790 297 824 399 730 764 178 918 901 56 438 353 241 365 574 991 848 57 803 484 731 148 706 647 333 380 952 191 630 604 305 695 187 282 469 487 33
4 801 77 446 226 396 692 429 963 892 721 461 363 308 805 44 591 98 680 36 740 785 608 420 204 405 45 979 413 103 806 826 812 274 496 825 677 373 129 221 743 226 176 897
 334 757 343 570 576 281 811 615 45 920 804 81 552 738 712 384 27 808 724 174 146 961 158 392 527 452 85 377 526 191 586 225 545 943 91 327 783 529 564 954 432 425 809
711 956 329 315 382 37 261 519 301 618 980 269 481 102 217 70 394 320 418 783 463 996 593 966 122 674 642 427 85 533 60 305 693 687 411 349 373 203 445 358 590 571 116
61 286 411 835 703 775 734 509 160 858 671 250 227 379 778 459 607 199 965 936 916 955 689 952 103 454 429 1 122 434 253 193 791 231 374 364 542 873 742 262 437 123 698
 347 0 763 992 21 409 648 790 287 833 969 666 424 120 112 371 153 794 304 440 770 507 666 200 210 484 46 385 286 517 56 493 900 661 17 506 640 625 16 289 231 729 288 99
4 498 505 300 669 477 999 216 508 377 744 337 775 670 136 736 0 925 865 227 385 581 619 429 650 828 302 744 959 705 612 7 41 888 618 128 876 973 768 730 201 44 599 452
669 392 746 695 826 955 505 828 684 414 530 600 278 137 48 39 742 301 897 999 79 272 306 175 294 457 310 255 791 76 423 145 200 144 900 217 630 191 947 746 151 310 617
209 384 86 496 965 800 498 712 983 212 835 746 638 140 632 704 438 686 209 825 714 906 45 660 680 318 955 194 333 763 998 585 944 255 933 998 819 508 594 987 687 840 49
```

```
---MergeSort---

El arreglo ordenado es: 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 6 6 6 6 7 7 7 7 7 7 8 8 8 8 9 9 9 9 9 9 9 10 10 10 11 11 11
11 11 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 14 14 14 14 14 15 15 15 15 15 15 16 16 17 17 17 18 18 18 18 18 18 18 18 18 19 19 19 19 19 20 20 21 21 21 21 21 22 22 22 22 22
23 23 24 24 24 24 24 25 25 26 26 26 27 27 27 27 27 27 28 28 28 28 28 28 29 29 29 29 30 30 30 30 31 31 31 31 31 31 31 31 31 32 33 33 33 33 34 34 34 34 34 34 34 35 35 35 36 36
36 36 36 37 37 37 37 37 37 38 39 39 39 39 39 39 39 40 40 40 40 40 40 41 41 41 42 43 43 43 44 44 44 44 44 44 44 44 44 44 44 44 44 45 45 45 45 45 45 45 45 46 46 46 46 46
46 46 46 47 47 47 47 47 47 47 48 48 48 48 48 49 49 49 49 49 49 50 50 50 51 51 51 52 52 52 52 53 53 53 53 54 54 54 54 55 55 55 55 56 56 56 56 57 57 57 57 57 57 58 58 58
58 59 59 59 59 59 59 60 60 60 61 61 61 61 61 61 62 62 62 62 62 62 63 63 63 63 63 63 63 64 64 64 64 64 64 65 65 65 65 65 65 65 65 66 66 66 66 66 66 67 67 67 68 68 68 68
68 68 69 69 69 69 69 69 70 70 70 70 71 71 71 71 71 72 72 73 74 74 74 74 74 74 74 74 75 75 75 76 76 76 76 76 76 77 77 77 77 77 77 78 78 78 78 78 79 79 79 79 79 80 81
81 81 81 81 82 82 82 82 82 82 83 83 83 83 83 83 83 84 84 84 84 84 84 85 85 85 85 85 85 85 85 85 85 85 86 86 86 86 86 86 87 87 87 87 87 87 87 88 88 88 88 89 89 89 89 89
89 89 89 89 89 90 91 91 91 91 91 91 91 91 92 92 92 93 93 93 94 94 94 95 95 95 95 95 95 96 96 96 96 96 96 96 96 97 97 97 97 97 97 97 97 98 98 98 98 98 98 98 99 99 99 99
99 100 100 101 101 101 102 102 102 102 103 103 103 103 103 104 104 104 104 104 104 104 105 105 105 105 105 106 106 106 106 106 106 107 107 107 108 108 108 108 108 109 109 1
09 109 109 109 109 109 110 110 111 111 111 111 111 111 112 112 112 112 113 113 113 113 113 113 113 113 113 114 114 114 114 114 114 114 114 115 115 115 115 115 116 116 1
16 117 117 118 118 118 118 119 119 119 119 119 120 120 120 120 120 120 121 122 122 122 122 122 122 122 122 122 123 123 123 123 123 124 124 124 124 125 126 126 126 126 127 1
27 127 127 127 128 128 128 128 129 129 129 129 129 129 129 130 130 130 131 131 131 131 131 132 132 132 132 132 132 133 133 134 134 134 134 135 135 135 136 136 136 136 1
37 137 137 137 137 137 137 138 138 138 138 138 139 139 139 139 139 139 140 140 140 140 140 140 140 141 141 141 141 141 143 143 144 144 144 144 144 144 144 144 144 1
45 145 145 145 146 146 146 146 146 146 147 147 148 148 148 148 149 149 149 150 150 150 151 151 151 151 151 151 152 152 152 152 152 152 152 153 153 153 153 153 154 154 1
54 155 155 155 156 156 156 156 156 157 157 157 158 158 158 158 158 158 159 159 159 159 159 159 159 159 160 160 160 161 161 162 162 162 162 163 163 164 164 164 164 165 1
65 165 165 165 165 166 166 166 166 166 166 167 167 167 168 168 169 169 169 169 169 169 169 170 170 170 170 170 170 171 171 171 171 171 171 172 172 172 172 173 173 173 173 173 1
73 173 174 174 174 174 174 174 174 174 175 175 176 176 177 177 177 177 177 177 177 177 178 178 178 179 179 179 180 180 180 181 181 182 182 182 182 182 182 182 182 182 1
82 182 183 183 183 184 184 185 185 185 185 185 186 186 186 186 187 187 187 188 188 188 188 188 188 188 189 189 189 190 190 190 191 191 191 191 191 191 191 191 191 191 1
92 192 192 192 193 193 193 193 194 194 194 195 196 196 196 196 196 196 197 197 197 197 197 198 198 198 199 199 199 199 199 200 200 200 200 200 201 201 202 202 202 202 2
02 203 203 203 203 203 203 203 203 204 204 204 204 204 204 204 204 205 205 205 205 205 205 205 205 206 206 206 206 206 206 207 207 207 207 207 207 207 208 208 208 208 2
08 208 208 209 209 209 209 209 209 209 210 210 210 210 210 210 211 211 211 211 211 212 212 212 212 212 212 213 213 213 213 213 214 214 214 214 215 215 215 216 216 216 2
16 216 216 217 217 217 217 217 217 218 218 218 219 219 219 219 220 220 220 220 221 221 221 221 221 221 222 222 222 222 223 223 223 223 223 223 223 223 223 224 2
24 224 224 224 224 225 225 225 225 225 225 225 226 226 226 226 226 226 226 227 227 227 227 227 227 227 227 227 227 227 228 228 228 228 228 228 229 229 229 230 2
30 230 230 231 231 231 231 231 231 231 231 231 232 232 232 232 233 233 233 233 233 233 233 233 233 233 234 234 234 234 235 235 235 235 236 236 237 238 238 238 238 2
39 239 239 239 239 239 239 240 240 240 240 240 240 240 241 241 241 242 243 243 243 243 243 243 243 243 244 244 244 245 245 245 245 245 245 245 246 246 247 247 247 247 2
47 247 247 247 248 248 248 248 249 249 249 249 249 227 249 250 250 250 250 250 250 250 250 251 251 251 251 251 251 252 252 253 253 253 253 253 254 254 254 254 255 255 255 255 2
55 255 255 255 255 255 256 256 256 256 256 257 257 257 257 258 258 258 258 258 258 258 259 259 259 259 259 259 260 260 260 260 260 261 261 261 261 261 262 262 262 262 2
62 262 263 263 263 263 263 263 263 264 264 264 264 265 265 265 265 265 265 266 267 267 267 267 267 268 268 268 268 268 269 269 269 269 270 270 270 270 271 271 271 271 2
71 271 271 272 272 272 273 274 274 274 274 274 274 274 274 275 275 275 276 276 276 276 276 277 277 277 277 277 277 277 278 278 278 278 279 280 280 280 280 281 281 2
81 281 281 281 282 282 282 282 282 282 282 282 282 283 284 284 285 285 285 286 286 286 286 286 286 287 287 287 287 287 287 288 288 288 288 289 289 289 289 289 289 2
89 290 290 290 290 290 290 290 290 291 291 291 291 292 292 292 292 293 293 294 294 294 294 294 295 295 295 295 295 296 296 297 297 297 297 298 298 298 298 299 299 2
99 299 300 300 300 300 300 300 300 300 301 301 301 301 301 301 301 301 302 302 302 302 302 302 302 302 302 303 303 303 303 304 304 304 304 304 304 304 304 305 305 305 3
05 306 306 306 306 306 306 306 306 306 307 307 307 307 307 308 308 308 308 308 308 308 309 309 309 309 309 309 310 310 310 310 310 310 310 310 311 311 311 311 311 312 3
12 312 313 313 313 314 314 314 314 314 315 315 315 315 315 315 316 316 316 316 316 316 316 316 316 316 317 317 318 318 318 318 319 319 319 319 319 319 320 320 320 320 321 3
21 321 321 321 322 322 322 323 323 323 323 323 323 323 323 324 324 324 324 324 324 324 324 325 325 326 326 326 326 326 327 327 327 328 328 328 328 328 329 330 330 331 331 3
```





Num. Operaciones Realizadas: 404204

Una vez realizadas 5 pruebas en cada uno de los algoritmos para arreglos de tamaño 50,100,500,800,1000,2000,5000 y 10000 se obtuvieron los siguientes resultados.

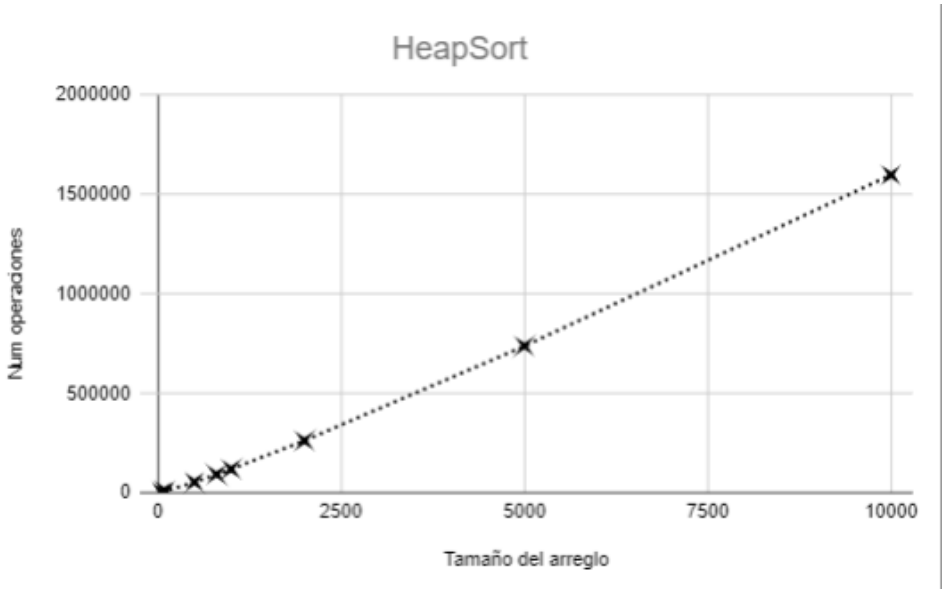
HeapSort

HeapSort								
Tamaño del arreglo	50	100	500	800	1000	2000	5000	10000
Prueba1(Num Operaciones)	3151	7762	52991	91708	117808	262337	737468	1598880
Prueba2 (Num operaciones)	3248	7657	53195	91818	118088	262365	739291	1599968
Prueba 3(Num Operaciones)	3155	7738	53073	91712	118474	260425	737349	1598463
Prueba4(Num Operaciones)	3305	7831	53186	91078	118690	261596	736851	1600170
Prueba5(Num operaciones)	3176	7852	52590	91880	117712	262351	738003	1598897
Promedio	3207	7768	53007	91639,2	118154,4	261814,8	737792,4	1599275,6

Núm. de Operaciones realizadas por cada tamaño de arreglo

Tamaño del arreglo	Num operaciones
50	3207
100	7768
500	53007
800	91639
1000	118154
2000	261815
5000	737792
10000	1599276

Promedio de Operaciones



Núm de Operaciones contra tamaño del arreglo

QuickSort

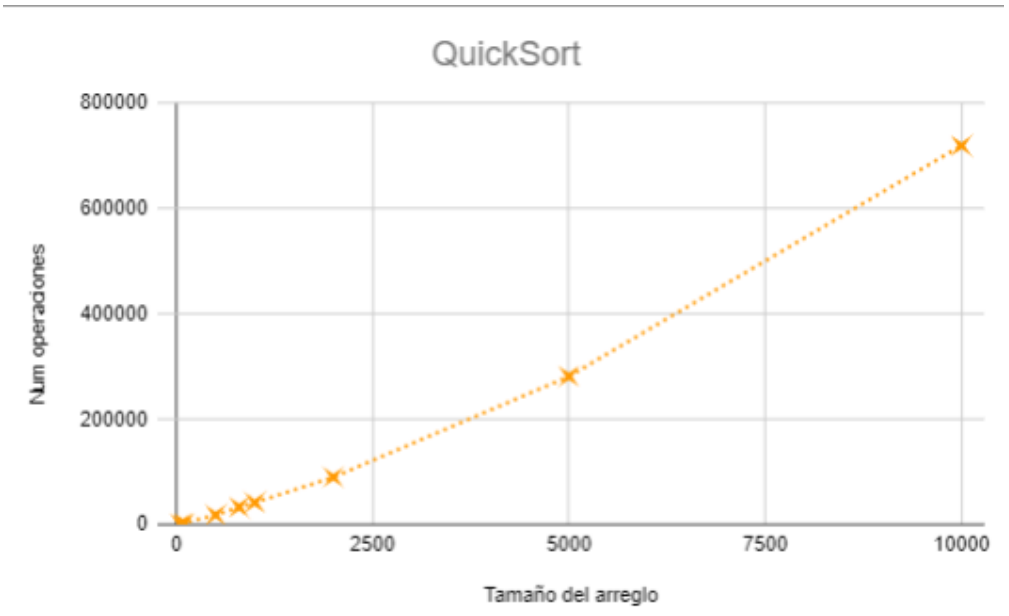
QuickSort

Tamaño del arreglo	50	100	500	800	1000	2000	5000	10000
Prueba1(Num Operaciones)	1072	2778	17142	35991	42122	92231	286305	827736
Prueba2 (Num operaciones)	1039	2557	15682	30833	40898	91950	271162	678158
Prueba 3(Num Operaciones)	1275	2448	21563	29074	41731	84490	297688	702642
Prueba4(Num Operaciones)	876	2349	17570	36045	40305	88820	269891	709499
Prueba5(Num operaciones)	960	3097	17231	32504	42501	91175	281907	682216
Promedio	1044,4	2645,8	17837,6	32889,4	41511,4	89733,2	281390,6	720050,2

Núm. de Operaciones realizadas por cada tamaño de arreglo

Tamaño del arreglo	Num operaciones
50	1044
100	2646
500	17838
800	32889
1000	41511
2000	89733
5000	281391
10000	720050

Promedio de Operaciones Realizada



Núm de Operaciones contra tamaño del arreglo

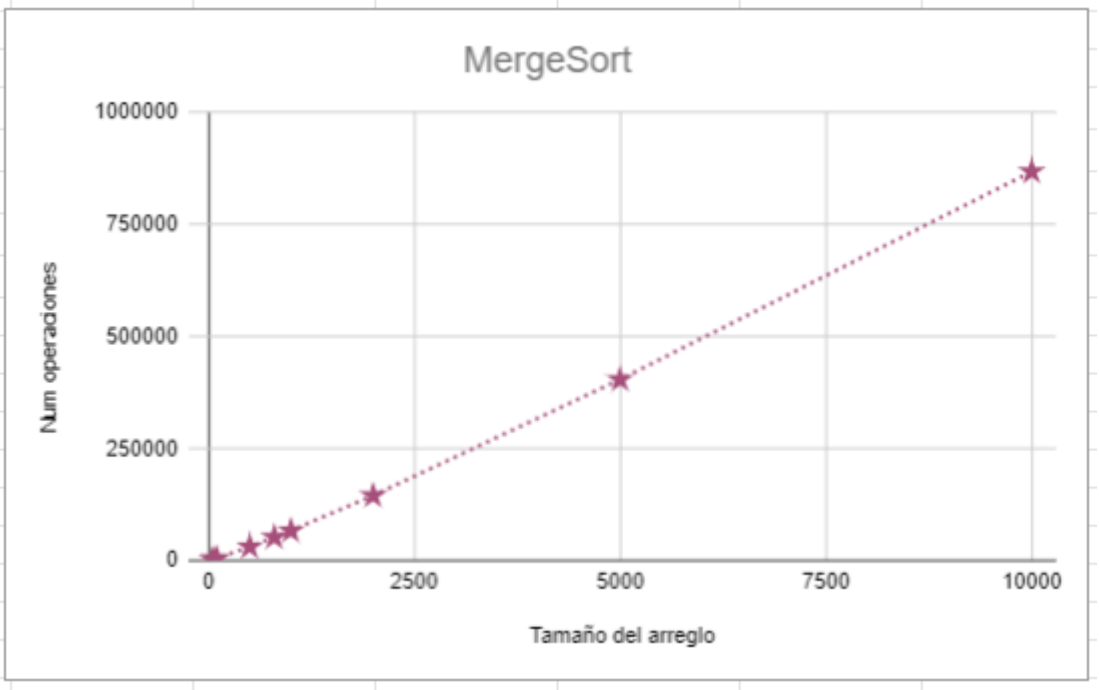
MergeSort

MergeSort								
Tamaño del arreglo	50	100	500	800	1000	2000	5000	10000
Prueba1(Num Operaciones)	2049	4694	30301	51988	66592	145177	404204	868593
Prueba2 (Num operaciones)	2054	4694	30275	51971	66595	145174	404245	868559
Prueba 3(Num Operaciones)	2051	4691	30309	52004	66587	145201	404260	868519
Prueba4(Num Operaciones)	2046	4695	30289	51984	66572	145177	404272	868447
Prueba5(Num operaciones)	2042	4699	30301	51971	66573	145142	404221	868540
Promedio	2048,4	4694,6	30295	51983,6	66583,8	145174,2	404240,4	868531,6

Núm. de Operaciones realizadas por cada tamaño de arreglo

Tamaño del arreglo	Num operaciones
50	2048
100	4695
500	30295
800	51984
1000	66584
2000	145174
5000	404240
10000	868532

Promedio de Operaciones Realizada



Tamaño de arreglo contra número de operaciones

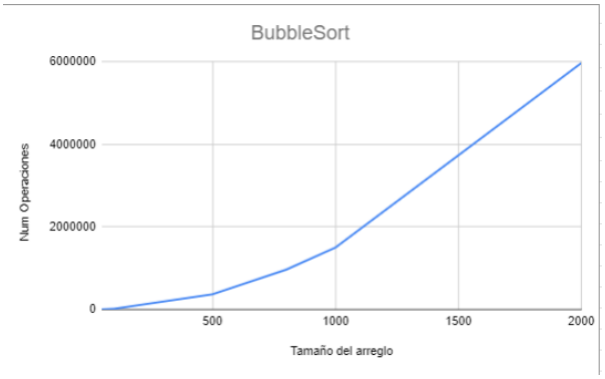
Cada uno de los algoritmos presentó el rendimiento esperado debido a su complejidad O(n log n) que gráficamente se observa como una función casi lineal en la cual no se percibe mucha curvatura a medida que incrementa el tamaño del arreglo.

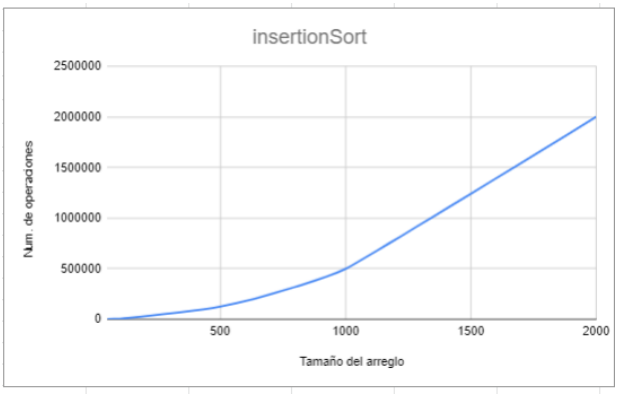
HeapSort es el algoritmo que realiza un mayor número de operaciones esto debido a que en su proceso tiene que “transformar” el arreglo a la estructura de un Max-Heap además de que como antes ocupa memoria adicional es por ellos que en comparación de merge y quick presenta más desventajas para casos promedios.

En MergeSort se pudo observar que a comparación de HeapSort y QuickSort el rango de operaciones que realiza en casos promedio es muy constante inclusive para arreglos de 10000 ya que la variación de sus operaciones en las ejecuciones realizadas van de 19 (568,540 - 568,559) hasta 146(868, 593 - 868, 447) esto puede ser consecuencia que para cada tamaño de arreglo hace las mismas particiones sin importar su contenido esto hace que en principio arreglos de 50 tengan el mismo número de operaciones y estas van a variar en el momento en que se realiza la mezcla que al ir mezclando pequeños bloques generas cada vez más bloques ordenados, por ende la variación de operaciones no es mucha.

Para QuickSort el rango de variación de las operaciones es mayor ya que toma como pivote el último elemento del arreglo y lo acomoda, genera sub arreglos y repite el procedimiento hasta que ya no pueda dividirse más y el número de operaciones que ejecuta para dividir los arreglos y el número de elementos en dichos sub arreglos no es constante ya que todo se maneja en torno al pivote.

Si comparamos las gráficas obtenidas de estos algoritmos contra las gráficas de bubbleSort, InsertionSort y selection observamos cómo en tan solo un arreglo de tamaño 2000 se observa un importante crecimiento en el número de operaciones pudiendo presenciar una curva con forma cuadrática siendo que el número de operaciones de HeapSort,QuickSort y MergeSort para arreglos de 10,000 elementos sigue siendo aún menor que el número de operaciones de Bubble,Insertion y selection para arreglos de 2000 elementos.





5. Ordenamiento en Java y practicando programas sencillos

Para QuickSort

El ejercicio fue realizado de forma exitosa, para dar un mejor formato y observar el procedimiento tanto de QuickSort como de merge se agrego el método printSubArray en la clase Utilidades para ir observando la modificación del arreglo

En cuanto a la inicialización del arreglo se decidió preguntar al usuario por los números mediante el método Inicializar ubicado de igual manera en utilidades

```
public class Principal {
    public static void main(String args[]) {
        //Declara e inicializa un arreglo de 20 elementos
        int[] arr = new int[20];
        Utilidades.Inicializar(arr);
        System.out.println("Ordenamiento QuickSort");
        QuickSort.quickSort(arr, 0, 19);
        System.out.println("El arreglo ordenado es: ");
        Utilidades.printArray(arr);
    }
}
```

Capturas de la Ejecución

```
Ingresa un numero: 777
Ingresa un numero: 1
Ingresa un numero: 2
Ingresa un numero: 69
Ingresa un numero: 33
Ingresa un numero: 27
Ingresa un numero: 105
Ingresa un numero: 109
Ingresa un numero: 666
Ingresa un numero: 17
Ingresa un numero: 10
Ingresa un numero: 25
Ingresa un numero: 27
Ingresa un numero: 30
Ingresa un numero: 78
Ingresa un numero: 88
Ingresa un numero: 34
Ingresa un numero: 128
Ingresa un numero: 333
Ingresa un numero: 21
```

```
Ordenamiento QuickSort
El pivote es: 21
Una vez colocado el pivote en su posicion
1 2 17 10 21 27 105 109 666 777 69 25 27 30 78 88 34 128 333 33
Izquierda
Sub Array:  1 2 17 10
El pivote es: 10
Una vez colocado el pivote en su posicion
1 2 10 17 21 27 105 109 666 777 69 25 27 30 78 88 34 128 333 33
Izquierda
Sub Array:  1 2
El pivote es: 2
Una vez colocado el pivote en su posicion
1 2 10 17 21 27 105 109 666 777 69 25 27 30 78 88 34 128 333 33
Izquierda
Sub Array:  1
Derecha
Sub Array:
Derecha
Sub Array:  17
Derecha
Sub Array:  27 105 109 666 777 69 25 27 30 78 88 34 128 333 33
El pivote es: 33
Una vez colocado el pivote en su posicion
1 2 10 17 21 27 25 27 30 33 69 105 109 666 78 88 34 128 333 777
Izquierda
Sub Array:  27 25 27 30
```

```
El arreglo ordenado es:
1 2 10 17 21 25 27 27 30 33 34 69 78 88 105 109 128 333 666 777
```

Para MergeSort

```
public class Principal {
    public static void main(String args[]) {
        //Declara e inicializa un arreglo de 20 elementos
        int[] arr = new int[20];
        Utilidades.Inicializar(arr);
        System.out.println("Ordenamiento MergeSort");
        MergeSort.sort(arr,0,19);
        System.out.println("El arreglo ordenado es: ");
        Utilidades.printArray(arr);
    }
}
```

```
Ingresa un numero: 444
Ingresa un numero: 1
Ingresa un numero: 4
Ingresa un numero: 567
Ingresa un numero: 777
Ingresa un numero: 45
Ingresa un numero: 10
Ingresa un numero: 27
Ingresa un numero: 23
Ingresa un numero: 29
Ingresa un numero: 66
Ingresa un numero: 669
Ingresa un numero: 333
Ingresa un numero: 45
Ingresa un numero: 76
Ingresa un numero: 45
Ingresa un numero: 98
Ingresa un numero: 99
Ingresa un numero: 123
Ingresa un numero: 345
Ordenamiento MergeSort
El arreglo ordenado es:
1 4 10 23 27 29 45 45 45 66 76 98 99 123 333 345 444 567 669 777
```

Para este ejercicio se me complicó el hecho de tener varios archivos de clases ya que en un principio al ir copiando los respectivos códigos dentro del proyecto deshabilite la opción de generar un método principal y al no generar el método principal no me di cuenta que el package no se coloca y cuando creí tener todo bien me generaba un error de ejecución, esto porque no había un package -que por cierto ya me estaba desesperando -. Me gusta java y me gusta el entorno de NetBeans pero al igual con C siento que hace falta mas familiarización en el entorno con el cual se trabaja.

Conclusiones

En la realización de la práctica se cumplieron con los objetivos ya que gracias a conceptos previos como complejidad computacional, punteros, funciones y paso de parámetros se probó el funcionamiento de los Algoritmos de QuickSort, MergeSort y HeapSort estudiados en clase, además, gracias a la implementación de un contador de operaciones realizadas, observamos de mejor manera el comportamiento de dichos algoritmos permitiendo analizar que su gráfica es muy parecida a una función lineal.

De igual forma se realizaron comparaciones con respecto a los algoritmos realizados de la primera práctica en donde el número de operaciones incrementó de forma desmedida con respecto al tamaño del arreglo, aunque si analizamos pequeños arreglos de 50 elementos y comparamos, los algoritmos de (n^2) presentan un rendimiento parecido a los algoritmos de complejidad O(n log n)

Específicamente el ejercicio 3 me ayudó a comprender de mejor manera el funcionamiento tanto de merge como de quick ya que se implementó las impresiones en pantalla de la forma en que va modificándose nuestro arreglo, pues al tratarse de algoritmos recursivos se me dificulta la parte de visualizar el cambio del arreglo en las iteraciones, por lo cual creo que dicho ejercicio debería de presentarse antes del 2 que es donde explicamos las comparaciones de los quickSort´s proporcionados y es justo la parte que me costó más explicar.Pero sobre todo me gustó que volviera a aparecer un ejercicio en java y que este sea más de implementación.

Me encontré esta imagen mientras hacía la practica xD