

Prácticas de Algorítmica.
3º de Grado en Ingeniería Informática.
Curso 2015-2016.
Práctica 3.

Objetivos.

Con esta práctica se pretende que el alumno implemente un algoritmo basado en la técnica divide y vencerás. Se proponen varios ejercicios de distinto nivel y el alumno ha de seleccionar uno de ellos. La puntuación máxima que se podrá obtener en cada uno de los ejercicios figura en su enunciado.

Enunciado 1:

Implementar el algoritmo recursivo para obtener el máximo y el mínimo elemento de un vector. El vector se rellenará aleatoriamente usando la función de la primera práctica. En la salida ha de aparecer el máximo y el mínimo elemento del vector así como el número de llamadas recursivas generadas. **Puntuación máxima = 8.**

Enunciado 2:

Implementar el algoritmo, basado en el quicksort, para obtener los k -menores elementos de un vector. El vector se rellenará aleatoriamente usando la función de la primera práctica. Implementar una función de prueba que verifique que el programa funcione correctamente. En la salida del programa se preguntará al usuario si se desea mostrar el vector resultante, resaltando los k -menores. **Puntuación máxima = 8.**

Enunciado 3:

Implementar el algoritmo de ordenación quicksort, para un vector de n elementos de tipo entero, usando las 3 opciones siguientes:

1. Usando el método de ordenación por inserción cuando el subvector a ordenar, resultante en las sucesivas iteraciones del quicksort, tenga un número de elementos menor o igual a χ , siendo χ un valor introducido por el usuario.
2. Usando el cálculo de la mediana de 5 elementos seleccionados aleatoriamente, para determinar el pivote en el algoritmo de partición. Esta forma de seleccionar el pivote se hará siempre y cuando el número de elementos del subvector a tratar sea igual o superior a un límite χ dado por el usuario.
3. Usando la implementación del quicksort dada en clase.

El usuario introducirá, al igual que en la práctica 1, el número mínimo y máximo de elementos del vector y el incremento, para determinar las pruebas a realizar (no hay que hacer repeticiones). El vector se generará aleatoriamente (elementos de tipo entero comprendidos entre 1 y 1000000) y después se introducirá el límite referenciado en los apartados 1 y 2. Posteriormente se ordenará el vector obtenido para cada valor de n usando las tres opciones (se han de hacer tres copias del vector original), para usar en todos los casos el mismo vector). El programa ha de mostrar gráficamente la evolución de los tiempos en función de n y de los valores de χ introducidos en los apartados 1 y 2. Para ello se fijará un valor de n y se hará variar χ . Para cada valor de n se generará una gráfica. Probar como mínimo tres valores de n superiores a 100.000.

Realizad distintas pruebas para valores de n y de χ y hacer un informe por escrito que refleje las conclusiones que habéis sacado de las pruebas. **Puntuación máxima = 10.**

Enunciado 4:

Implementar el algoritmo de ordenación por fusión, para un vector de n elementos de tipo entero usando el método de ordenación por inserción cuando el subvector a ordenar, resultante en las sucesivas iteraciones del método, tenga un número de elementos menor o igual a χ , siendo χ un valor introducido por el usuario. Usad la función que implementasteis en la práctica 1 para verificar que el vector está ordenado.

El usuario introducirá, al igual que en la práctica 1, el número mínimo y máximo de elementos del vector y el incremento, para determinar las pruebas a realizar (no hay que hacer repeticiones). El vector se generará

aleatoriamente (elementos de tipo entero comprendidos entre 1 y 1000000) y después se introducirá el valor de χ . Posteriormente se ordenará el vector obtenido para cada valor de n . El programa ha de mostrar gráficamente la evolución de los tiempos en función de n y del valor de χ . Para ello se fijará un valor de n y se hará variar χ . Para cada valor de n se generará una gráfica. Probar como mínimo tres valores de n superiores a 100.000.

Realizad distintas pruebas para valores de n y de χ y hacer un informe por escrito que refleje las conclusiones que habéis sacado de las pruebas. **Puntuación máxima = 10.**

Enunciado 5:

Implementar el algoritmo para multiplicar enteros grandes. Para ello, el alumno ha de implementar una clase **Entero** que ha de tener sobrecargados el operador $*$, el operador $+$ (es necesario para el producto) y los operadores de entrada/salida ($>>$ y $<<$). El número de dígitos máximo pueden tener dos números para multiplicarse de forma directa lo introducirá el usuario. El profesor suministrará una serie de algoritmos implementados en C y que deberán ser adaptados dentro de la clase **Entero**, que facilitarán la implementación de dicha clase.

El programa tendrá un menú en el que se seleccione la operación que se desea realizar (suma o producto) y en cada operación se pedirán los números a utilizar. **Puntuación máxima = 10.**

Fecha de comienzo: 22 de Octubre de 2015.

Fecha máxima de entrega: 5 de Noviembre de 2015.