

Lab 1: Algoritmos de ordenación

NOMBRE: DARIO FERNANDO VINUEZA MIER

Tiempo asignado para el laboratorio 2 horas. El trabajo realizado se evalúa al acabar el laboratorio y la entrega se realizará en la fecha indicada en el sistema de entrega de actividades.

Descripción del laboratorio

En el laboratorio vamos a realizar un análisis empírico de los algoritmos de ordenación.

Para medir el tiempo de ejecución y número de operaciones básicas de los algoritmos tenemos que instrumentalizar el código. Para ello:

1. Antes y después de llamar a una función de ordenación llamamos a `clock()` que nos devuelve el tiempo consumido por nuestro proceso. La ventaja de usar `clock()` es que solo cuenta el tiempo consumido por nuestro proceso, si hay otros procesos ejecutándose en paralelo, no se cuenta el tiempo de los demás procesos.
2. Creamos una variable contador global (`BasicOpCounter`) que se pone a cero antes de llamar a un algoritmo de ordenación, se incrementa cada vez que se ejecuta la operación básica, y se lee su valor al acabar el algoritmo de ordenación.

Implemente estas funciones en un fichero `solution.c`.

Entrega del laboratorio

Una vez acabado el trabajo, adjunte `solution.c` donde se implementen las funciones de `solution.h`.

Adjunte un fichero `respuestas.docx` donde se responde a estas preguntas:

a) ¿Funciona bien tu programa?, ¿Qué fallos existen?

No tengo ningún fallo, ni tampoco presenta ninguna advertencia, por lo que podría afirmar que el programa esta funcionando de forma correcta.

Adjunto captura de pantalla del programa:

```

147 do {
148     while (i < j && dictionary[i] < splitter_value)
149         i++;
150     while (j > i && dictionary[j] >= splitter_value)
151         j--;
152     if (i < j) {
153         BasicOpCounter++;
154         char *tmp;
155         tmp = dictionary[i];
156         dictionary[i] = dictionary[j];
157         dictionary[j] = tmp;
158     }
159     while (i < j) {
160         if (j > left) {
161             BasicOpCounter++;
162             char *tmp;
163             tmp = dictionary[left];
164             dictionary[left] = dictionary[j];
165             dictionary[j] = tmp;
166         }
167     }
168     //printf("B0: %llu\n", BasicOpCounter);
169     return j;
170 }
171
172 void quick_sort(char* dictionary[], int left, int right) {
173     if (left < right) {
174         int i = split(dictionary, left, right);
175         quick_sort(dictionary, left, i-1);
176         quick_sort(dictionary, i+1, right);
177     }
178 }
179
180
181
182
183
184

```

b) Rellena la siguiente tabla con las fórmulas que estiman el número de operaciones básicas.

Análisis matemático		
Algoritmo	Formula cerrada número medio de operaciones básicas	Número estimado de operaciones básicas para $n=800$, $n=8000$ y $n=80000$
Burbuja	<ul style="list-style-type: none"> N: 800 $OB = 319600$ $Tiempo = 0.002344 \text{ seg}$ N: 8000 $OB = 31996000$ $Tiempo = 0.247505 \text{ seg}$ N: 80000 $OB = 3199960000$ $Tiempo = 28.280383 \text{ seg}$ 	$C_{avg}(n) = \frac{1}{2} \cdot \frac{n(n-1)}{2}$ <ul style="list-style-type: none"> N: 800 $C_{avg}(n) = 159800$ N: 8000 $C_{avg}(n) = 15998000$ N: 80000 $C_{avg}(n) = 1599980000$
Selección	<ul style="list-style-type: none"> N: 800 $OB = 319600$ $Tiempo = 0.001301 \text{ seg}$ N: 8000 $OB = 31996000$ $Tiempo = 0.126859 \text{ seg}$ 	$C_{avg}(n) = \frac{1}{2} n \cdot (n-1)$ <ul style="list-style-type: none"> N: 800 $C_{avg}(n) = 319600$ N: 8000

	<ul style="list-style-type: none"> N: 80000 <p>$OB = 3199960000$ $Tiempo = 12.608336 \text{ seg}$</p>	<p>$C_{avg}(n) = 31996000$</p> <ul style="list-style-type: none"> N: 80000 <p>$C_{avg}(n) = 3199960000$</p>
Inserción	<ul style="list-style-type: none"> N: 800 <p>$OB = 799$ $Tiempo = 0.000647 \text{ seg}$</p> <ul style="list-style-type: none"> N: 8000 <p>$OB = 7999$ $Tiempo = 0.066004 \text{ seg}$</p> <ul style="list-style-type: none"> N: 80000 <p>$OB = 79999$ $Tiempo = 8.624804 \text{ seg}$</p>	<p>$C_{avg}(n) = \frac{1}{2} \cdot \frac{n(n-1)}{2}$</p> <ul style="list-style-type: none"> N: 800 <p>$C_{avg}(n) = 159800$</p> <ul style="list-style-type: none"> N: 8000 <p>$C_{avg}(n) = 15998000$</p> <ul style="list-style-type: none"> N: 80000 <p>$C_{avg}(n) = 1599980000$</p>
Mergesort	<ul style="list-style-type: none"> N: 800 <p>$OB = 7776$ $Tiempo = 0.001921 \text{ seg}$</p> <ul style="list-style-type: none"> N: 8000 <p>$OB = 103808$ $Tiempo = 0.018013 \text{ seg}$</p> <ul style="list-style-type: none"> N: 80000 <p>$OB = 1308928$ $Tiempo = 0.203363 \text{ seg}$</p>	<p>$C_{avg}(n) = n \cdot \log_2 n$</p> <ul style="list-style-type: none"> N: 800 <p>$C_{avg}(n) = 7715.08$</p> <ul style="list-style-type: none"> N: 8000 <p>$C_{avg}(n) = 103726.27$</p> <ul style="list-style-type: none"> N: 80000 <p>$C_{avg}(n) = 1303016.99$</p>
Quicksort	<ul style="list-style-type: none"> N: 800 <p>$OB = 798$ $Tiempo = 0.000258 \text{ seg}$</p> <ul style="list-style-type: none"> N: 8000 <p>$OB = 8458$ $Tiempo = 0.041399 \text{ seg}$</p> <ul style="list-style-type: none"> N: 80000 <p>$OB = 20971$ $Tiempo = 0.292654 \text{ seg}$</p>	<p>$C_{avg}(n) = n \cdot \log_2 n$</p> <ul style="list-style-type: none"> N: 800 <p>$C_{avg}(n) = 7715.08$</p> <ul style="list-style-type: none"> N: 8000 <p>$C_{avg}(n) = 103726.27$</p> <ul style="list-style-type: none"> N: 80000 <p>$C_{avg}(n) = 1303016.99$</p>

c) Son los resultados empíricos obtenidos consistentes con los teóricos

No necesariamente en todos los casos, los resultados obtenidos presentan el siguiente resumen.

- **Burbuja:** Se puede observar que el valor obtenido duplica al valor teórico.
- **Selección:** En este caso los valores son los mismos, para ambos resultados.
- **Inserción:** Los valores no son iguales, presentan una aplica diferencia.
- **Mergesort:** En este caso los valores son parcialmente similares, presentan una alta aproximación.

- **Quicksort:** En este caso los valores son parcialmente similares, presentan una alta aproximación.

d) Si existen diferencias entre los tiempos de ejecución experimentales y teóricos indica cuáles son y a qué crees que se deben.

Como se menciona en la pregunta anterior, se puede observar que existe una alta diferencia con respecto al método de **Inserción**.

Una de las causas puede ser la capacidad de la maquina en la que se esta ejecutando el programa, lenguaje de programación, recursos que usa el sistema operativo.

Adjunto las características de la maquina donde se realizaron las pruebas solicitadas para este laboratorio.

