Escuela de Ingeniería Informática
Universidad de Oviedo

Algoritmia
2016-2017

Control 1 – 6 de marzo de 2017

Apellidos, nombre _____ NIF: _____

Pregunta 1 (1 p.)

Responde a las siguientes preguntas:

a) (0,5p.) Consideremos el algoritmo Quicksort con pivote central aplicado sobre vectores aleatorios. Si dicho algoritmo toma 27 microsegundos para n=512, calcula el tiempo que tardará para n=1024.

b) (0,5p.) Considere ahora un algoritmo con complejidad $O(n^2)$. Si para t = 43 microsegundos el método pudiera resolver un problema con un tamaño de n = 512, ¿cuál podría ser el tamaño del problema si dispusiéramos de un tiempo de 172 microsegundos?

Pregunta 2 (2 p.)

Indica la complejidad temporal de los siguientes fragmentos de código (escríbela directamente en esta hoja de examen):

a) (0,5 p.)

```
public void metodo1(int n) {
            int t = 200;
            for (int i = 2*n; i>=0; i -= 3) {
                  for (int j = i; j <= n*n*n; j*=4) {
                        System.out.println("Secuencia");
                        t++;
                        for (int k=0; k<n; k *= 2) {
                              System.out.println(t);
                        }
                  }
            }
      }
```

b) (0,5 p.)

```
   public void metodo2(int n, int p) {
         if (n < 0)
               sum= 1;
         else {
               sum = 0;
               metodo2(n/2, p);
               metodo2(n/2, p);
               for (int i = 0; i<n; i++) {
                     for (int j = 0; j<n; j++) {
                           sum++;
                     }
               }
               System.out.println("sum:" + sum);
         }
   }
```

c) (1 p.) Escribir un método recursivo en java que simule una función Divide y Vencerás por división con una complejidad $O(n^2 \log n)$ y el número de subproblemas = 4.

Pregunta 3 (3 p.)

Por favor, responde a las siguientes preguntas:

a) (1 p.) Dada la siguiente secuencia de números: **60, 40, 20, 10, 30, 50, 70** ordénalos utilizando Quicksort con la estrategia **del elemento central** para seleccionar el pivote en cada iteración. Indica claramente la traza del algoritmo marcando el pivote escogido en cada paso y las particiones creadas.

b) (0,5 p.) Explica la complejidad del algoritmo para los casos peor, mejor y medio e indica cuando se dan estos casos.

c) (0,75 p.) Dada la siguiente secuencia de números: **60, 40, 20, 10, 30, 50, 70** ordénalos utilizando Mergesort. Indica claramente la traza del algoritmo, marcando el orden en el que se realizan las diferentes llamadas recursivas y la combinación de los diferentes subvectores para alcanzar el vector ordenado final.

d) (0,25 p.) Explica cuál es la complejidad de dicho algoritmo.

e) (0,5 p.) ¿Cuáles son las diferencias y similitudes entre los algoritmos de ordenación Quicksort y Mergesort?

Pregunta 4 (2 p.)

Convertir la implementación del algoritmo que suma los elementos de un vector DV recursivo a una implementación que utilice hilos para ejecutar cada una de las llamadas de forma paralela.

```
public class SumaVector
{
	static int []v;

	public static int suma(int[]v)
	{
		return recDiv (0,v.length-1);
	}

	private static int recDiv(int iz,int de)
	{
		if (iz==de)
			return v[iz];
		else
		{
			int m=(iz+de)/2;
			return recDiv(iz,m)+recDiv(m+1,de);
		}
	}

	public static void main(String arg [] )
	{
		int n = 100;
		v=new int[n];
		for ( int i=0; i<n;i++) v[i]=i;
		System.out.println ("SOLUCION ="+suma(v));
	}
}
```

Pregunta 5 (2 p.)

En muchas entrevistas de trabajo, las grandes empresas piden a los candidatos resolver un problema algorítmico para pasar una prueba. Un ejemplo es el siguiente:

Se parte de una colección de números ordenados, donde cada número aparece exactamente dos veces (excepto un número que aparecerá solamente en una ocasión). Diseña un algoritmo **altamente eficiente** para encontrar el valor que aparece solo.

a) (0,5 p.) Explica el algoritmo (en lenguaje natural o pseudocódigo es suficiente) y por qué éste mejora a otras soluciones.
b) (0,5 p.) Razona la complejidad del algoritmo.
c) (1 p.) Aplica el algoritmo a los siguientes casos:

```
5

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

10 10 17 17 18 18 19 19 21 21 23

1 3 3 5 5 7 7 8 8 9 9 10 10
```

Si las soluciones no son claras o no están bien explicadas, se darán por incorrectas, por lo que se recomienda explicar muy claramente tanto el algoritmo como cada paso realizado.