

Programación II - 2010/2011 - Universitat Jaume I

Evaluación continua - Módulo 1 de teoría - Recuperación

17 de mayo de 2011

La duración máxima de esta prueba es de 45 minutos. No puedes consultar libros ni apuntes.

Ejercicio 1 (3,5 puntos)

Analiza el tiempo de ejecución del siguiente programa y exprésalo empleando la notación O . Indica cuál es el tiempo de ejecución total de cada uno de los bucles que aparecen numerados y del programa completo.

```
import java.util.Scanner;

public class EnigmaCostes {
    public static boolean enigma(int [] v, int d) {
        int a = 0, b = v.length - 1;
        while (a <= b) { // Indica el coste. BUCLE 1 -----
            int c = (a + b) / 2;
            if (v[c] == d)
                return true;
            else if (v[c] < d)
                a = c + 1;
            else
                b = c - 1;
        }
        return false;
    }
    public static int contar(int [] v1, int[] v2) {
        int x = 0;
        for (int i = 0; i < v1.length; i++) // Indica el coste. BUCLE 2 -----
            if ( enigma(v2, v1[i]) )
                x++;
        return x;
    }
    public static void main(String [] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.print("Dígame un número entero positivo: ");
        int n = entrada.nextInt();
        int [][] m = new int[n][n];

        for (int i = 0; i < n; i++) // Indica el coste. BUCLE 3 -----
            for (int j = 0; j < n; j++)
                m[i][j] = i * j;

        int cont = 0;
        for (int i = 1; i < n * n; i++) // Indica el coste. BUCLE 4 -----
            if (enigma( m[n - 1], i))
                cont++;

        for (int i = 1; i < n; i++) // Indica el coste. BUCLE 5 -----
            for (int j = i + 1; j < n; j++)
                cont += contar(m[i], m[j]);

        System.out.println(cont);
    }
    // Indica el coste del programa completo: -----
}
```

Ejercicio 2 (3 puntos)

Escribe en lenguaje Java un método que reciba como argumentos dos vectores de enteros $v1$ y $v2$. Suponiendo que los vectores que se le pasan están siempre ordenados de menor a mayor y que ninguno de ellos contiene elementos repetidos, el método debe contar cuántos elementos de $v1$ están en $v2$ y devolver el resultado.

El tiempo de ejecución de tu solución debe ser $O(n)$, siendo n la cantidad total de elementos en ambos vectores.

Ejercicio 3 (3,5 puntos)

Considera el siguiente programa:

```
public class MisterioRecurción {

    public static boolean misterio(String c) {
        return misterio(c, 0, c.length() - 1);
    }

    public static boolean misterio(String c, int a, int b) {

        System.out.println("a=" + a + " b=" + b);
        if ( a >= b )
            return true;
        if ( c.charAt(a) != c.charAt(b) )
            return false;

        return misterio(c, a+1, b-1);
    }

    public static void main(String [] args) {
        String [] vectorCadenas = {"abba", "ababba", "reponer", "reconocer"};

        for (int i = 0; i < vectorCadenas.length; i++) {
            System.out.println( vectorCadenas[i] );
            System.out.println( misterio(vectorCadenas[i]) );
        }
    }
}
```

1. Escribe todo lo que se muestra en la salida estándar al ejecutarlo, en el orden en que aparece en la salida.
Observa bien que hay una llamada a `System.out.println` en el segundo método `misterio`.
2. Describe brevemente para qué sirve el método `misterio`.
3. Expresa empleando la notación O el tiempo de ejecución del método `misterio`.