

ESTRUCTURA DE DATOS (2016-2017)
SUBGRUPO C2
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Relación 1: Eficiencia

Mario Rodríguez Ruiz

6 de noviembre de 2016

Índice

1	Problema 1.3	3
2	Problema 1.6	3
3	Problema 1.9	3
4	Problema 1.10	5
5	Problema 1.11	6

1. Problema 1.3

Ordenar de menor a mayor los siguientes órdenes de eficiencia:

$$n, \sqrt{n}, n^3 + 1, n^2, n \log_2(n^2), n \log_2 \log_2(n^2), 3^{\log_2(n)}, 3^n, 2^n, 2^n + 3^{n-1}, 20000, n + 100, n 2^n$$
$$n \log_2 \log_2(n^2) < n \log_2(n^2) < 3^{\log_2(n)} < \sqrt{n} < n < n^2 < 2^n < 2^n + 3^{n-1} < \\ < n 2^n < n^3 + 1 < 3^n < n + 100 < 20000$$

2. Problema 1.6

Demostrar que si $f(n) \ni O(g(n))$ y $g(n) \ni O(h(n))$ entonces $f(n) \ni O(h(n))$

$$O(n) = \{1, 10, 5, \log_2(n), \log_7(n), 3\log_2(n), \sqrt{n}, n, \dots\}$$

$$O(n^2) = \{n^2, \dots, 3n^2 + n, \dots\} \cup O(n)$$

$$f(n) = \log_2(n) \ni O(\log_2(n))$$

$$f(n) = \log_2(n) \ni O(n)$$

$$g(n) = n^2 \ni O(n^2)$$

3. Problema 1.9

Obtener usando la notación O-mayúscula la eficiencia del siguiente trozo de código:

```
1 for (int i=0; i<n; i++)
2   for (int j=0; j<n; j++){
3     C[i][j]= 0;
4     for (k=0; k<n; k++)
5       C[i][j] += A[i][k] * B[k][j];
6   }
```

- Línea 1: 3 OE (asignación, comparación, incremento)
- Línea 2: 3 OE (asignación, comparación, incremento)
- Línea 3: 2 OE (acceso al elemento C[i][j], asignación)
- Línea 4: 3 OE (asignación, comparación, incremento)
- Línea 5: 7 OE (4 accesos a elementos de las matrices, 2 operaciones, asignación)

Se tienen tres bucles anidados, haciendo cada uno **n iteraciones**. El resto de la función es de orden constante, por lo que puede identificarse cada una como **O(1)**. De esta forma, el tiempo de ejecución en función de n es:

$$T(n) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n 1 = \sum_{i=0}^n \sum_{j=0}^n n = n^3 \quad (3.1)$$

Por lo que se puede confirmar que la eficiencia del código es de $O(n^3)$ (3.2)

4. Problema 1.10

Obtener usando la notación O-mayúscula la eficiencia de la siguiente función:

```

1 void ejemplo(int n)
2 {
3     int i, j, k;
4     for (i = 1; i < n; i++)
5         for (j = i+1; j <= n; j++)
6             for (k = 1; k <= j; k++)
7                 Global += k*i;
8 }
```

- Línea 4: 3 OE (asignación, comparación, incremento)
- Línea 5: 4 OE (asignación, comparación, 2 incrementos)
- Línea 6: 4 OE (asignación, comparación, incremento)
- Línea 7: 3 OE (asignación, 2 operaciones)

Bucle de la j

$$Tj(n) = \frac{(n-i) * (1+n-i)}{2} \rightarrow O(n^2) \quad (4.1)$$

Bucle de la i

$$Ti(n) = \sum_{i=0}^n \left(\frac{(n-i) * (1+n-i)}{2} \right) \quad (4.2)$$

Por lo que el orden de eficiencia de la función es de $O(n^2)$ (4.3)

5. Problema 1.11

Obtener usando la notación O-mayúscula la eficiencia del siguiente trozo de código:

```
1  for (i=0; i<n; i++)
2      if (i%2){
3          for (j=i; j<n; j++)
4              x*=i ;
5          for (j=1; j<i; j++)
6              y*=j ;
7      }
```

- Línea 1: 3 OE (asignación, comparación, incremento)
- Línea 2: 1 OE (operación)
- Línea 3: 3 OE (asignación, comparación, incremento)
- Línea 4: 2 OE (producto, asignación)
- Línea 5: 3 OE (asignación, comparación, incremento)
- Línea 6: 2 OE (producto, asignación)

Bucle de la j

$$\sum_{j=i}^n 1 = n - i \quad (5.1)$$

$$\sum_{j=1}^i 1 = i \quad (5.2)$$

$$Tj(n) = n - i + i = n \quad (5.3)$$

$$T(n) = \sum_{i=0}^n n = n \sum_{i=0}^{\frac{n}{2}} 1 = n \frac{n}{2} = \frac{n^2}{2} \quad (5.4)$$

$$\text{Por lo que se puede confirmar que la eficiencia del código es de } O(n^2) \quad (5.5)$$