



## Examen de Algorítmica.

Curso 2014–2015. Convocatoria ordinaria de Junio. Duración: 3 horas.

### Grado en Ingeniería Informática.

#### 1. (2.5 pt) Eficiencia

- Escribir la ecuación de recurrencia para la complejidad en tiempo correspondiente a la siguiente función (m potencia de 2)

```
unsigned char septiembre (int i, int j, int m)
{
    if (m==1)
        if (i==j)
            return 1;
        else return 0;

    for (k=1;k<=m;k++)
        if (septiembre(i,k,m/8) && septiembre (k,j,m/8))
            return 1;
        else return 0;
}
```

- 3 Resolver la recurrencia siguiente:

$$T(n) = 2 \times T(\sqrt{n}) + \log_2 \log_2 n \quad n \geq 4 \quad T(2)=1$$

#### 2. (2 pt.) Divide y Vencerás

- Dado un vector de enteros (positivos y negativos) de tamaño  $n$ , encontrar el subvector de suma máxima, esto es, se debe cumplir que la suma de todos los elementos del subvector sea la mayor posible.

#### 3. (1.5 pt.) Greedy

- Encontrar (usando el algoritmo de Dijkstra) el camino más corto entre el vértice 1 y todos los demás en el grafo con la siguiente matriz de adyacencias:

	1	2	3	4
1	0	4	3	9
2	2	0	7	3
3	7	9	0	8
4	5	8	5	0

#### 4. (1.5 pt.) Programación Dinámica

- Obtener la ecuación de recurrencia para el problema de las 3 mochilas 0/1 resuelto mediante **Programación Dinámica**. En este caso los objetos no se pueden partir en trozos, y hay un total de  $n$  objetos distintos cada uno con un peso  $p_i$  y un beneficio  $b_i$ . Suponemos que hay un total de tres mochilas disponibles, cuyas capacidades máximas respectivas son  $M1$ ,  $M2$  y  $M3$ . El objetivo es llenar las tres mochilas maximizando el beneficio de los objetos transportados (Indicación: una ecuación de recurrencia permite definir una sola función mochila, aunque esta puede tener más parámetros de entrada).

#### 5. (2.5 pt.) Branch and bound

- Tenemos un conjunto de barcos ( $b1, b2, b3, b4$ ) y un conjunto de pasajeros ( $p1, p2, p3, p4$ ). El coste de que un determinado conjunto de pasajeros  $p_i$  viaje en el barco  $b_j$  viene dado por la siguiente tabla:

	b1	b2	b3	b4
p1	95	2	55	69
p2	75	11	89	83
p3	63	89	9	77
p4	12	75	82	22

Diseñar un algoritmo para asignar los pasajeros a los barcos con el mínimo coste global.