

## Metodología de la Programación y Algoritmia

Convocatoria de Junio 2018

Puntuación Total: 4.6

NºExpediente: \_\_\_\_\_

DNI: \_\_\_\_\_

Apellidos: \_\_\_\_\_

Nombre: \_\_\_\_\_

1.- Juan va a un restaurante y pide la carta. En la carta se indica junto con el nombre de cada plato el número de kilocalorías (kcal) que tiene. Juan quiere pedir un conjunto de platos (sin repetir plato) para comer, de forma que la suma de las kilocalorías de los platos que elija sea lo más aproximada a la cantidad máxima de kilocalorías que le ha recomendado su dietista sin sobrepasar dicha cantidad. Hay que tener en cuenta que como Juan es muy comiente cuando selecciona un plato para comer se lo come completo, con lo cual adquiere el total de kilocalorías del plato.

Se quiere diseñar un algoritmo RyP para determinar qué platos debe elegir Juan de forma que cumpla todos los requisitos que se especifican.

1.a) (0.2 puntos) Tipifica el problema.

1.b) (0.4 puntos) Define una función de cota y la estructura del nodo utilizando pseudocódigo. Indica qué significa cada una de las variables que contiene el nodo.

1.c) (1.1 puntos) Genera el árbol de expansión del problema suponiendo que la cantidad máxima que Juan debe consumir es 1600 kcal y que en la carta hay 4 platos que tienen 500 kcal, 600 kcal, 300 kcal y 600 kcal, respectivamente.

- Dentro de cada nodo indica los valores que tienen las variables que contiene.
- Numera a cada nodo según el orden en el que se ha generado.
- Indica el estado de la lista de nodos vivos en cada momento, cómo se va actualizando y el orden en el que se expanden los nodos. Los nodos se extraen según el valor de la función de cota que hayas definido en el apartado anterior y en el caso de que coincida aplica una estrategia FIFO.
- ¿Cuántos nodos se han generado, podado y expandido?
- Interpreta la solución final obtenida.

2.- Dado el siguiente algoritmo

```
función junio18(a:natural U {0}, b:natural U {0}):natural U {0}
  si a < b
    devolver b + junio18(a + 1, b - a)
  si no
    devolver a * b
  fsi
ffunción
```

2.a) (0.5 puntos) Realiza la traza de llamadas recursivas para a=3 y b=20 e indica cuál es el resultado final.

2.b) (1.0 puntos) Obtén la versión iterativa aplicando el esquema general más adecuado para este tipo de recursividad.

3.- (1.4 puntos) Calcula la complejidad asintótica del siguiente algoritmo y **justifica** tu respuesta de forma que quede claro por qué sale dicha complejidad. La complejidad asintótica de calcular(A:real[n]):real es  $O(n^2 \log n)$ .

```
función junio18(X:real[t]):real
(1)   i,j:entero
(2)   k,n:real
(3)   i ← 1
(4)   mientras i ≤ t hacer
(5)     j ← i
(6)     mientras j > 1 hacer
(7)       k ←  $X_j - t / 2$ 
(8)       j ← j - 2
(9)     fmientras
(10)    i ← i * 2
(11)   fmientras
(12)   n ← calcular(X) + t
(13)   devolver n + t
ffunción
```

**Duración:** 2 horas y 15 minutos.

Se entregan todas las hojas (incluida esta hoja). Escribe en todas las hojas tu nombre y apellidos.