

Parcial-1-2018-RESUELTO.pdf



un_saltador



Análisis y diseño de algoritmos



2º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**

EOI Escuela de
organización
industrial

La mejor escuela de negocios en
energía, sostenibilidad y medio
ambiente de España.

Más información
www.eoi.es

Formamos
talento para un futuro
Sostenible



100% Empleabilidad



Modalidad: Presencial u online



**Programa de Becas,
Bonificaciones y Descuentos**

ELIMINA IMPERFECCIONES

RÁPIDA ABSORCIÓN

NO DEJA RESIDUOS

Todo lo que **tu piel**
necesita cuando menos
te lo esperas.



TU RUTINA ANTI-IMPERFECCIONES

GARNIER
PureActive

E. T. S. de Ingeniería Informática
Análisis y Diseño de Algoritmos

Cursos 2018-19
Control Bloque 1

Alumno: _____ Grado: _____

Ejercicio 1

Especificar un programa que dado un vector de enteros v , con $v.length \geq 0$, lo positivice. Positivizar un vector consiste en sustituir en el vector todas las apariciones de números negativos por 0, dejando sin modificar el resto de posiciones.

Ejercicio 2

Dada la ecuación en recurrencia $T(n) = 2T(n-1) + n$, con $T(0) = 0$, se pide resolver por el polinomio característico y dar su complejidad.

Ejercicio 3

Se dispone de un array con el peso de n bolas, donde $n = 3^k$. Una de estas bolas es más ligera que el resto y se desea encontrarla, pero no podemos acceder a la información del array de forma directa. Para ello, se dispone de una función que nos da el peso de un grupo de bolas, pero solo puede utilizarse $2k$ veces. Diseñar e implementar un algoritmo de Divide y Vencerás que realice la tarea de encontrar esa bola. ¿Cuál es la complejidad del algoritmo diseñado? Plantea la ecuación de recurrencia y utiliza el teorema maestro.

Nota 1: Asumir que existe ya implementada una función `sum(int a[], int start, int end)` que devuelve la suma de los pesos de las bolas desde la posición `start` a la posición `end`.

Nota 2: Implementar el algoritmo con todo el nivel de detalle posible.

Ejercicio 4

Marca la respuesta correcta introduciendo entre paréntesis (V)erdadero o (F)also.

1. La clase NP podría resolverse en tiempo polinómico con un lenguaje de programación no determinista ()
2. Al transformar un problema de optimización a un problema de decisión, la complejidad cambia en un factor $\log(n)$ ()
3. Los problemas NP-completos son los más complejos dentro de la clase NP ()
4. Los problemas NP-hard son al menos tan complejos como los NP-completos ()
5. El problema de la satisfacibilidad se resuelve con algoritmos $O(2^n)$ de complejidad temporal y $O(n)$ de complejidad espacial ()
6. En la clase espacial L, el espacio es reutilizable, el tiempo no. En la clase espacial EXP el espacio no es reutilizable ()

Puntuaciones: E1: 6 puntos; E2: 7 puntos; E3: 14 puntos; E4: 3 puntos [1/2 punto acierto, -1/4

WUOLAH

Control Bloque I curso 2018-2019

① $P \equiv \{ l = v.length \geq 0, v_{ant} = v \}$.

positivizar (int[] v) : void;

$$Q \equiv \{ \forall i \in [0, l) / v_{ant}[i] < 0 \Rightarrow v[i] = 0 \wedge \\ \wedge \forall i \in [0, l) / v_{ant}[i] \geq 0 \Rightarrow v_{ant}[i] = v[i] \}.$$

② $T(n) = 2T(n-1) + n, T(0) = 0.$

$$T(n) - 2T(n-1) = n1^n \Rightarrow (x-2)(x-1)^2 = 0.$$

$$\Rightarrow T(n) = \alpha 2^n + \beta + \gamma n.$$

$$T(0) = 0, T(1) = 2 \cdot T(0) + 1 = 1, T(2) = 2 \cdot T(1) + 2 = 4.$$

$$\Rightarrow \left. \begin{array}{l} \alpha + \beta = 0 \\ 2\alpha + \beta + \gamma = 1 \\ 4\alpha + \beta + 2\gamma = 4 \end{array} \right\}$$

$$\left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 \\ 4 & 1 & 2 & 4 \end{array} \right) \sim \left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & -3 & 2 & 4 \end{array} \right) \sim \left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{array} \right)$$

$$\gamma = -1, \beta = -2, \alpha = 2$$

$$\Rightarrow T(n) = 2 \cdot 2^n - 2 - n \in \Theta(2^n).$$

**BEST
LIFE**
EXPERIENCE

Duerme bajo las estrellas

Del Sahara



Marruecos, Aventura en el Sahara

en 5 días



- ♥ Transporte privado en autobús
- ♥ Pasajes de **barco** ida y vuelta
- ♥ Visita a **medina de Fez, Chefchaoun y Assilah**
- ♥ **Guía Marroquí** durante todo el viaje
- ♥ Excursión **en camellos** por las dunas del sahara
- ♥ Atardecer y amanecer en **el desierto**
- ♥ Alojamiento 4 noches, media pensión (**Hotel + Jaima**)
- ♥ **Fotos** del viaje



Análisis y diseño de algoritmos



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Banco de apuntes de la

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



③ $n = 3^k$. A lo sumo, 2k veces la función sum.

```
public static int posBola(int[] v, int inf, int sup,
double suma) {
    if (suma == 0) suma = sum(v, inf, sup);
    if (sup <= inf) return sup;
    else if (inf + 1 == sup) return (v[inf] < v[sup] ? inf : sup);
    else {
        int m = (inf + sup) / 2;
        double sumParcial = sum(v, inf, m);
        if (sumParcial < (suma * (m - inf + 1) / (sup - inf + 1))) {
            return posBola(v, inf, m, suma - (suma * (sup - m) / (sup - inf + 1)));
        } else {
            return posBola(v, m + 1, sup, suma - (suma * (m - inf + 1) / (sup - inf + 1)));
        }
    }
}
```

Aclaraciones:

- La primera llamada a posBola debe ser con los siguientes parámetros:

$\text{posBola}(v, 0, v.\text{length}-1, 0);$

- Tome la suma como double porque trabajo con fracciones, y si pierdo decimales podría fallar.

Complejidad: $T(n) = T(n/2) + n + k$, $k \in \mathbb{N}$ es una constante.

ELIMINA IMPERFECCIONES
RÁPIDA ABSORCIÓN
NO DEJA RESIDUOS

Todo lo que **tu piel**
necesita cuando menos
te lo esperas.



TU RUTINA ANTI-IMPERFECCIONES

GARNIER
PureActive

Por el teorema maestro, como tenemos que $\log_2 1 = 0 < 1$, donde el último 1 es el exponente de n , deducimos que nuestro algoritmo es de orden n .

Además, nuestro algoritmo sum se cuenta $1 + \lfloor \log_2 k \rfloor$ veces, pues $1 + \lfloor \log_2 k \rfloor \leq 1 + \log_2 k \leq \leq 2k$. Demostremos la última desigualdad.

Sea $f(x) = 2x - \log_2 x - 1$, $f: (0, +\infty) \rightarrow \mathbb{R}$.

$f'(x) = 2 - \frac{1}{\ln 2} \cdot \frac{1}{x}$, pero como $k \in \mathbb{N}$ (con el cero) $\Rightarrow \frac{1}{k \cdot \ln 2} \leq \frac{1}{\ln 2} < 2 \Rightarrow 2 - \frac{1}{\ln 2} > 0$,

luego $\forall k \geq 1$, f es estrictamente creciente, y como $f(1) = 1$, tenemos que $1 \leq f(x) \forall x \geq 1$, luego

$f(x) \geq 0 \forall x \geq 1 \Rightarrow 2x - \log_2 x - 1 \geq 0 \forall x \geq 1 \Rightarrow$

$\Rightarrow 2x \geq \log_2 x + 1$, por lo que siempre usaremos la función sum un número permitido de veces.

④ 1) falso, no existen lenguajes de programación no deterministas. (Son palabras textuales del profesor, aunque yo pienso que es cierto, pues dice "podría", y adjunto un texto al final de este PDF con la explicación.

2) No le sabe el profesor, ni yo tampoco. Si alguno lo sabe, que me avise por favor. Yo pienso que podría ser en un faulto polinómico, pero ni idea ii.

WUOLAH

3) V. Por definición.

4) V. También por definición.

5) V.

6) F. El espacio siempre es reutilizable.

Complejidad computacional. Problemas P y NP

- Cualquier problema de optimización puede ser transformado en un problema de decisión. En el análisis de complejidad se manejan, por lo tanto, los problemas de decisión, que incluyen a los de optimización.
- Si se encuentra solución a un problema de decisión, entonces se encuentra también solución a un problema de optimización. Al estudiar los problemas de decisión, se pueden encontrar varias clases:
 - La clase de **problemas P** está formada por todos aquellos problemas de decisión para los cuales se tiene un algoritmo de solución que se ejecuta en **tiempo polinomial en una máquina determinista**.
 - La clase de **problemas NP** está formado por todos aquellos problemas de decisión para los cuales existe un algoritmo de solución que se ejecuta en **tiempo polinomial en una (hipotética) máquina no determinista**. Dicho de otro modo, **no se ha encontrado un algoritmo determinista que lo resuelva en tiempo polinomial**.



-32% DE MARCAS

FPS 50+
CADA DIA

NO GRASO

Sabemos que la única luz que te da
últimamente es la del flexo, pero está
guay que para **cuando acaben los
parciales tu cara esté perfecta**



TU RUTINA ANTI -IMPERFECCIONES

GARNIER
PureActive

WUOLAH



WUOLAH

Duerme bajo las estrellas del Sahara ¡Clic aquí!

**BEST
LIFE**
EXPERIENCE

Viajar nunca fue tan

Barato



Marruecos

en 3 días

- ♥ Transporte privado en autobús
- ♥ Pasajes de Barco ida y vuelta
- ♥ Visitas a Chefchaouen, Assilah, Tánger y Tetuán
- ♥ Guías nativos profesionales durante todo el viaje
- ♥ Pequeño paseo en camello
- ♥ 2 Noches en Hotel de 3 o 4 estrellas
- ♥ Desayunos y cenas incluidos
- ♥ Fotos del viaje

