

# EDA-2019-FEBTeoria-1.pdf



Anónimo



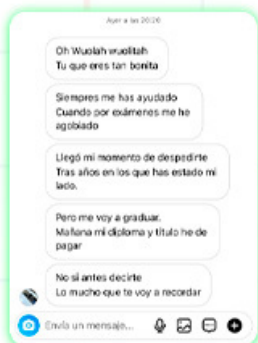
Estructura de Datos y Algoritmos



2º Grado en Desarrollo de Videojuegos



Facultad de Informática  
Universidad Complutense de Madrid



**Que no te escriban poemas de amor  
cuando terminen la carrera**



*(a nosotros por  
suerte nos pasa)*

**WUOLAH**

# WUOLAH

Oh Wuolah wuolilah  
Tu que eres tan bonita

--

3. (0.25 pt) En la técnica algorítmica de «vuelta atrás», ¿qué son los marcadores y para qué sirven? Pon un pequeño ejemplo.

4. (0.25 pt) Indica los costes de los métodos `empty`, `search` y `remove` de los árboles binarios de búsqueda (BST).

5. (0.25 pt) Obtén la recurrencia de coste  $T(n)$  de la siguiente función recursiva.

```

1 int f3(unsigned int n) {
2     if (n == 0)
3         return 0;
4     else
5         return f3(n-1) + f3(n-1) + n;
6 }
```

6. (1.25 pt) A partir de la siguiente recurrencia  $T(n)$  que representa el coste de un algoritmo recursivo, utiliza el método de las expansiones para calcular en qué orden de complejidad está incluida  $T(n)$ . Se deben realizar **todos los pasos** e indicar claramente el resultado obtenido en cada uno de ellos.

$$T(n) = \begin{cases} 8 & \text{si } n = 0 \\ 3T(n-1) + 5 & \text{si } n > 0 \end{cases}$$

#### Recordatorio

$\sum_{i=a}^n [f(i) + g(i)] = \sum_{i=a}^n f(i) + \sum_{i=a}^n g(i)$	$\sum_{i=a}^n k \cdot s_i = k \cdot \sum_{i=a}^n s_i$	$\sum_{i=a}^n i = \frac{(a+n)(n-a+1)}{2}$	$\sum_{i=0}^{n-1} k^i = \frac{1-k^n}{1-k}$
--	---	---	--