

Comenzado el lunes, 20 de marzo de 2023, 09:15**Estado** Finalizado**Finalizado en** lunes, 20 de marzo de 2023, 09:37**Tiempo empleado** 21 minutos 28 segundos**Puntos** 4,50/12,00**Calificación** 3,75 de 10,00 (37,5%)**Pregunta 1**

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Cuál de los siguientes algoritmos de ordenación necesita un espacio de almacenamiento adicional al vector que se ordena con complejidad $O(n)$?

Seleccione una:

- ☐ a. Quicksort.
- ☒ b. No contesto (equivalente a no marcar nada). ❌
- ☐ c. Mergesort.
- ☐ d. Bubblesort.

Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

De las siguientes expresiones, o bien dos son verdaderas y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta a las otras dos.

Seleccione una:

- ☐ a. $O(4^{\log_2(n)}) \subseteq O(n^2) \subset O(2^n)$
- ☐ b. $O(2^{\log_2(n)}) \subseteq O(n^2) \subset O(n!)$
- ☒ c. $O(n^2) \subset O(2^{\log_2(n)}) \subset O(2^n)$ ✔️
- ☐ d. No contesto (equivalente a no marcar nada).

Pregunta 3

Correcta

Se puntúa 1,00 sobre 1,00

Tenemos un vector desordenado y queremos obtener los tres elementos más pequeños. ¿Cuál sería la complejidad temporal más ajustada para hacerlo?

(sin pérdida de generalidad puedes suponer que en el vector todos los elementos son distintos)

Seleccione una:

- ☒ a. Lineal con la longitud del vector ✔️
- ☐ b. El logaritmo de la longitud del vector
- ☐ c. Cuadrática con la longitud del vector
- ☐ d. No contesto (equivalente a no marcar nada).

Pregunta 4

Correcta

Se puntúa 1,00 sobre 1,00

Con respecto al parámetro n , ¿Cuál es la complejidad temporal de la siguiente función?

```
void f( unsigned n ){
    if( n < 2 ) return;
    for( int i = 0; i < pow(n,2); i++ )
        cout << " ";
    for( int i = 0; i < 5; i++ )
        f( n / 2 );
}
```

Seleccione una:

- ☐ a. $\Theta(n^2 \log n)$
- ☐ b. $\Theta(n^2)$
- ☒ c. $\Theta(5^{\log n})$ ✓
- ☐ d. No contesto (equivalente a no marcar nada).

Pregunta 5

Incorrecta

Se puntúa -0,50 sobre 1,00

Las siguientes funciones calculan el valor de la potencia n -ésima de dos. ¿Cuál es más eficiente en cuanto a coste temporal?

```
unsigned long pot2_1(unsigned n){
    if (n==0) return 1;
    if (n%2==0) return pot2_1( n/2 ) * pot2_1( n/2 );
    else return 2 * pot2_1( n/2 ) * pot2_1( n/2 );
}
```

```
unsigned long pot2_2(unsigned n){
    if (n==0) return 1;
    unsigned long aux= pot2_2( n/2 );
    if (n%2==0) return aux * aux;
    else return 2 * aux * aux;
}
```

Seleccione una:

- ☐ a. La primera, pot2_1(n), es más eficiente que la otra.
- ☒ b. Las dos funciones son equivalentes en cuanto a coste temporal. ✗
- ☐ c. La segunda, pot2_2(n), es más eficiente que la otra.
- ☐ d. No contesto (equivalente a no marcar nada).

Pregunta 6

Incorrecta

Se puntúa 0,00 sobre 1,00

Si $f \in \Omega(g_1)$ y $f \in \Omega(g_2)$ entonces

Seleccione una:

- ☐ a. $f \in \Omega(g_1 \cdot g_2)$
- ☐ b. $f \in \Omega(g_1 + g_2)$
- ☒ c. No contesto (equivalente a no marcar nada). ✗
- ☐ d. $f \notin \Omega(\min(g_1, g_2))$

Pregunta 7

Correcta

Se puntúa 1,00 sobre 1,00

La complejidad temporal (o coste temporal asintótico) en el mejor de los casos...

Seleccione una:

- ☒ a. ... es una función de la talla, o tamaño del problema, que tiene que estar definida para todos los posibles valores de ésta. ✓
- ☐ b. ... es el tiempo que tarda el algoritmo en resolver la talla más pequeña que se le puede presentar.
- ☐ c. No contesto (equivalente a no marcar nada).
- ☐ d. Las otras dos opciones son ambas verdaderas.

Pregunta 8

Incorrecta

Se puntúa -0,50 sobre 1,00

La siguiente relación de recurrencia expresa la complejidad de un algoritmo recursivo, donde $g(n)$ es una función polinómica:

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Di cuál de las siguientes afirmaciones es falsa:

Seleccione una:

- ☐ a. Si $g(n) \in \Theta(1)$ la relación de recurrencia representa la complejidad temporal del algoritmo de búsqueda dicotómica.
- ☐ b. No contesto (equivalente a no marcar nada).
- ☐ c. Si $g(n) \in \Theta(n)$ la relación de recurrencia representa la complejidad temporal del algoritmo de ordenación Mergesort.
- ☒ d. Si $g(n) \in \Theta(n)$ la relación de recurrencia representa la complejidad temporal en el caso mejor del algoritmo de ordenación Quicksort. ✗

Pregunta 9

Correcta

Se puntúa 1,00 sobre 1,00

Los algoritmos de ordenación Quicksort y Mergesort tienen en común ...

Seleccione una:

- ☒ a. ... que aplican la estrategia de divide y vencerás. ✓
- ☐ b. ... que se ejecutan en tiempo $O(n)$.
- ☐ c. No contesto (equivalente a no marcar nada).
- ☐ d. ... que ordenan el vector sin usar espacio adicional.

Pregunta 10

Incorrecta

Se puntúa -0,50 sobre 1,00

Se pretende obtener la complejidad temporal en el caso más desfavorable de la siguiente función.

```
int exa (vector < int > & v){
    int i, sum = 0, n = v.size();
    if (n > 0){
        int j = n;
        while (sum < 100 and j != 0 ){
            j = j / 2;
            sum = 0;
            for (i = j; i < n; i++)
                sum += v[i];
        }
        return j;
    }
    else return -1;
}
```

¿Cuál de las siguientes formulaciones expresa mejor dicho coste?

Seleccione una:

- ☐ a. $c_s(n) = \sum_{j=0}^{n/2} \left(\frac{1}{2} \sum_{i=j}^n 1 \right) \in O(n \log n)$
- ☐ b. $c_s(n) = n \sum_{j=1}^{\log n} \sum_{i=1}^j \left(\frac{1}{2} \right)^i \in O(n \log n)$
- ☐ c. No contesto (equivalente a no marcar nada).
- ☒ d. $c_s(n) = \sum_{j=0}^{n/2} \sum_{i=j}^n 1 \in O(n^2)$ ✖

Pregunta 11

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Cuál de las formulaciones expresa mejor el coste temporal de la siguiente función?

```
int f(int n){
    int count = 0;
    for (int i = 2; i < n; i += 2)
        for (int j = 1; j < i; j*=2)
            count += 1;
    return count;
}
```

Seleccione una:

- ☒ a. No contesto (equivalente a no marcar nada). ✖
- ☐ b. $C_e(n) = \frac{\sum_{i=2}^{n-1} \log_2 2i}{2}$
- ☐ c. $C_e(n) = \sum_{i=1}^{n/2} (1 + \log_2 i)$
- ☐ d. $C_e(n) = \sum_{i=1}^{n/2} (\log_2(n/2))$

Pregunta 12

Correcta

Se puntúa 1,00 sobre 1,00

Sea la siguiente relación de recurrencia:

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 8T(\frac{n}{8}) + g(n) & \text{en otro caso} \end{cases}$$

Si $T(n) \in \Theta(n^2)$, ¿en cuál de estos tres casos nos podemos encontrar?

Seleccione una:

- ☐ a. $g(n) = n^3$
- ☒ b. $g(n) = n^2$ ✓
- ☐ c. $g(n) = n$
- ☐ d. No contesto (equivalente a no marcar nada).

[◀ Entrenamiento_primer_parcial](#)[Segundo parcial ▶](#)