

Primer parcial

Comenzado el	martes, 26 de marzo de 2013, 19:50
Completado el	martes, 26 de marzo de 2013, 20:08
Tiempo empleado	18 minutos 47 segundos

Pregunta 1 Correcta Puntúa como 1,00 🚩 Marcar pregunta	Sea $f(n)$ la solución de la relación de recurrencia $f(n) = 2f(n/2) + n$; $f(1) = 1$. Indica cuál de estas Seleccione una: <input type="radio"/> a. $f(n) \in \Theta(n^2)$ <input type="radio"/> b. $f(n) \in \Theta(n)$ <input checked="" type="radio"/> c. $f(n) \in \Theta(n \log(n))$ ✓
--	--

Pregunta 2 Correcta Puntúa como 1,00 🚩 Marcar pregunta	Para que la complejidad de un algoritmo presente caso mejor y peor distintos ... Seleccione una: <input type="radio"/> a. ... es condición necesaria y suficiente que existan instancias distintas del problema con el mismo tamaño. <input checked="" type="radio"/> b. ... es condición necesaria que existan instancias distintas del problema con el mismo tamaño. ✓ <input type="radio"/> c. ... es condición suficiente que existan instancias distintas del problema con el mismo tamaño.
--	--

Pregunta 3 Correcta Puntúa como 1,00 🚩 Marcar pregunta	Indica cuál es la complejidad, en función de n , del fragmento siguiente: <pre>int a = 0; for(int i = 0; i < n; i++) for(int j = i; j > 0; j /=2) a += A[i][j];</pre> Seleccione una: <input type="radio"/> a. $O(n)$ <input checked="" type="radio"/> b. $O(n \log n)$ ✓ <input type="radio"/> c. $O(n^2)$
--	--

Pregunta 4 Correcta Puntúa como 1,00 🚩 Marcar pregunta	Un problema de tamaño n puede transformarse en tiempo $O(n)$ en siete de tamaño $n/7$; por otro lado, la solución al problema cuando la talla es 1 requiere un tiempo constante. ¿cual de estas clases de coste temporal asintótico es la más ajustada? Seleccione una: <input type="radio"/> a. $O(n^2)$ <input type="radio"/> b. $O(n)$ <input checked="" type="radio"/> c. $O(n \log n)$ ✓
--	---

Pregunta 5

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Dada la siguiente relación de recurrencia, ¿Qué cota es verdadera?

$$f(n) = \begin{cases} 1 & n=1 \\ \sqrt{n} + 3f(n/3) & n>1 \end{cases}$$

Seleccione una:

- ☐ a. $f(n) \in \Theta(n^3)$
- ☒ b. $f(n) \in \Theta(n)$ ✓
- ☐ c. $f(n) \in \Theta(\sqrt{n} \log n)$

Pregunta 6

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Indica cuál es la complejidad, en función de n , del siguiente fragmento de código:

```
s=0; for(i=0;i<n;i++) for(j=i;j<n;j++) s+=i*j;
```

Seleccione una:

- ☐ a. $O(n^2)$ pero no $\Omega(n^2)$.
- ☒ b. $\Theta(n^2)$ ✓
- ☐ c. $\Theta(n)$

Pregunta 7

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

¿Cuál de estas tres expresiones es falsa?

Seleccione una:

- ☐ a. $3n^2 + 1 \in O(n^3)$
- ☒ b. $n + n \log(n) \in \Theta(n)$ ✓
- ☐ c. $n + n \log(n) \in \Omega(n)$

Pregunta 8

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Un algoritmo recursivo basado en el esquema *divide y vencerás* ...

Seleccione una:

- ☐ a. Las demás opciones son verdaderas.
- ☐ b. ... nunca tendrá una complejidad exponencial.
- ☒ c. ... será más eficiente cuanto más equitativa sea la división en subproblemas. ✓

Pregunta 9

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

¿Cuál es la complejidad temporal de la siguiente función recursiva?

```
unsigned desperdicio (unsigned n){  
    if (n<=1)  
        return 0;  
    unsigned sum = desperdicio (n/2) + desperdicio (n/2) + desperdicio (n/2);  
    for (unsigned i=1; i<n-1; i++)  
        for (unsigned j=1; j<=i; j++)  
            sum+=i*j;  
    return sum;  
}
```

Seleccione una:

- ☐ a. $\Theta(n^2 \log n)$
- ☒ b. $\Theta(n^2)$ ✓
- ☐ c. $\Theta(3^n)$

Pregunta 10

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Indica cuál es la complejidad de la función siguiente:

```
unsigned sum( const mat &A ) {    // A es una matriz cuadrada
unsigned d = A.n_rows();
unsigned a = 0;
    for( unsigned i = 0; i < d; i++ )
        for( unsigned j = 0; j < d; j++ )
            a += A(i,j);
return a;
}
```

Seleccione una:

- ☐ a. $O(n^2)$
- ☐ b. $O(n \log n)$
- ☒ c. $O(n)$ ✓

Pregunta 11

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Indica cuál de estas tres expresiones es cierta:

Seleccione una:

- ☐ a. $O(n^2) \subset O(2^{\log(n)}) \subset O(2^n)$
- ☐ b. $O(n^2) \subset O(2^{\log(n)}) \subseteq O(2^n)$
- ☒ c. $O(2^{\log(n)}) \subset O(n^2) \subset O(2^n)$ ✓