

Pregunta 1

Correcta

Puntúa como 1,00

 Marcar pregunta

Un problema de tamaño n puede transformarse en tiempo $O(n^2)$ en nueve de tamaño $n/3$; 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:

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

Pregunta 2

Correcta

Puntúa como 1,00

 Marcar pregunta

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

```
k=n/4;
for( int i = k; i < n - k; i++){
    A[i] = 0;
    for( int j = i - k; j < i + k; j++ )
        A[i] += B[j];
}
```

Seleccione una:

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

Pregunta 3

Correcta

Puntúa como 1,00

 Marcar pregunta

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

Seleccione una:

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

Pregunta 4

Correcta

Puntúa como 1,00

 Marcar pregunta

La versión de *Quicksort* que utiliza como pivote el elemento del vector que ocupa la posición central ...

Seleccione una:

- ☒ a. ... se comporta mejor cuando el vector ya está ordenado. ✓
- ☐ b. ... se comporta peor cuando el vector ya está ordenado.
- ☐ c. ... no presenta casos mejor y peor distintos para instancias del mismo tamaño.

Pregunta 5

Correcta

Puntúa como 1,00

 Marcar pregunta

Sea $f(n)$ la solución de la relación de recurrencia $f(n) = 2f(n/2) + 1$; $f(1) = 1$. Indica cuál de estas tres expresiones es cierta:

Seleccione una:

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

Pregunta 6

Incorrecta

Puntúa como 1,00

 Marcar pregunta

Para que la complejidad de un algoritmo presente caso mejor y peor distintos ...

Seleccione una:

- ☒ a. ... es condición necesaria y suficiente que existan instancias distintas del problema con el mismo tamaño. ✗
- ☐ b. ... es condición necesaria que existan instancias distintas del problema con el mismo tamaño.
- ☐ c. ... es condición suficiente que existan instancias distintas del problema con el mismo

tamaño.

Pregunta 7

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Indica cuál es la complejidad en función de n , donde k es una constante (no depende de n), del fragmento siguiente :

```
for( int i = k; i < n - k; i++){  
    A[i] = 0;  
    for( int j = i - k; j < i + k; j++ )  
        A[i] += B[j];  
}
```

Seleccione una:

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

Pregunta 8

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Indica cuál de estas tres expresiones es cierta:

Seleccione una:

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

Pregunta 9

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Pertenece $3n^2 + 3$ a $O(n^3)$?

Seleccione una:

- ☐ a. No.
- ☐ b. Sólo para $c = 1$ y $n_0 = 5$.
- ☒ c. Sí. ✓

Pregunta 10

Incorrecta

Puntúa como 1,00

🚩 Marcar pregunta

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

$$f(n) = \begin{cases} 1 & n = 1 \\ n^2 + 3f(n/3) & n > 1 \end{cases}$$

Seleccione una:

- ☐ a. $f(n) \in \Theta(n)$
- ☐ b. $f(n) \in \Theta(n^2)$
- ☒ c. $f(n) \in \Theta(n^2 \log n)$ ✗

Pregunta 11

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

El coste temporal asintótico del fragmento

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

y el del fragmento

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

son ...

Seleccione una:

- ☒ a. ... iguales. ✓
- ☐ b. ... el del primero, menor que el del segundo.
- ☐ c. ... el del segundo, menor que el del primero.

Pregunta 12

Incorrecta

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);  
    for (unsigned i=1; i<n-1; i++)  
        for (unsigned j=1; j<=i; j++)  
            for (unsigned k=1; k<=j; k++)  
                sum+=i*j*k;  
    return sum;  
}
```

Seleccione una:

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