Pregunta 1

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

La complejidad temporal en el mejor de los casos...

Seleccione una:

- a. ... es el tiempo que tarda el algoritmo en resolver la talla más pequeña que se le puede presentar.
- b. Las demás opciones son verdaderas.
- c. ... es una función de la talla que tiene que estar definida para todos los posibles valores de ésta.

Pregunta 2

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

Sobre la complejidad temporal de la siguiente función:

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

Seleccione una:

- a. Ninguna de las otras dos alternativas es cierta.
- b. Las complejidades en los casos mejor y peor son distintas.
- c. El mejor de los casos se da cuando $n \leq 1$ y en tal caso la complejidad es constante.

Pregunta 3

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

Con respecto al esquema *Divide y vencerás*, ¿es cierta la siguiente afirmación?

Si la talla se reparte equitativamente entre los subproblemas, entonces la complejidad temporal resultante es una función logarítmica.

Seleccione una:

- a. No, nunca, puesto que también hay que añadir el coste de la división en subproblemas y la posterior combinación.
- b. No tiene porqué, la complejidad temporal no depende únicamente del tamaño resultante de los subproblemas.
- c. Sí, siempre, en Divide y Vencerás la complejidad temporal depende únicamente del tamaño de los subproblemas.

Pregunta 4

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

¿Qué cota se deduce de la siguiente relación de recurrencia?

$$f(n) = \begin{cases} 1 & n = 1 \\ n + 4f(n/2) & n > 1 \end{cases}$$

Seleccione una:

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

Pregunta 5

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

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

Seleccione una:

- a. $2n^3 - 10n^2 + 1 \in O(n^3)$
- b. $n + n\sqrt{n} \in \Omega(n)$
- c. $n + n\sqrt{n} \in \Theta(n)$

Pregunta 6

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

Sea $f(n) = n \log n + n$.

Seleccione una:

- a. ... $f(n) \in \Omega(n \log n)$
- b. ... $f(n) \in O(n \log n)$
- c. Las otras dos opciones son ciertas

Pregunta 7

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

Si $f_1(n) \in O(g_1(n))$ y $f_2(n) \in O(g_2(n))$ entonces...

Seleccione una:

- a. Las otras dos alternativas son ciertas.
- b. $f_1(n) + f_2(n) \in O(\max(g_1(n), g_2(n)))$
- c. $f_1(n) + f_2(n) \in O(g_1(n) + g_2(n))$

Pregunta 8

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

La versión de *Quicksort* que utiliza como pivote la mediana del vector...

Seleccione una:

- (x) a. ... no presenta caso mejor y peor distintos para instancias del mismo tamaño.
- ( ) b. ... es más eficiente si el vector ya está ordenado.
- ( ) c. ... es la versión con mejor complejidad en el mejor de los casos.

Pregunta 9

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

El siguiente fragmento del algoritmo de ordenación *Quicksort* reorganiza los elementos del vector para obtener una subsecuencia de elementos menores que el pivote y otra de mayores. Su complejidad temporal, con respecto al tamaño del vector `v`, que está delimitado por los valores `pi` y pf, es...

```
x = v[pi];
i = pi+1;
j = pf;
do {
      while( i<=pf && v[i] < x ) i++;
      while( v[j] > x ) j--;
      if( i <= j ) {
                     swap( v[i],v[j]);
                     i++;
                     j--;
      }
} while( i < j );
swap(v[pi],v[j]);
```

Nota: La función swap tiene una complejidad temporal constante.

Seleccione una:

- a. ... lineal en cualquier caso.
- b. ... cuadrática en el peor de los casos.
- c. ... lineal en el caso peor y constante en el caso mejor.

Pregunta 10

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

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

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

Seleccione una:

- a. $f(n) \in \Omega(2^n)$
- b. $f(n) \in \Theta(n^2)$
- c. $f(n) \in \Theta(2^n)$

Pregunta 11

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

¿Cual es la solución a la siguiente relación de recurrencia?

$$f(n) = \begin{cases} \Theta(1) & n = 0 \\ \Theta(1) + f(n/3) & n > 0 \end{cases}$$

Seleccione una:

- a. $f(n) \in \Theta(\log n)$.
- b. $f(n) \in \Theta(n/3)$.
- c. Ninguna de las otras dos es cierta.

Pregunta 12

Respuesta guardada

Puntúa como 1,00

Marcar pregunta

En cuanto a la complejidad temporal de la siguiente función:

```
int ejemplo (vector < int > & v){

    int n=v.size();
    int j,i=2;
    int sum=0;
    while (n>0 && i<n){
        j=i;
        while (v[j] != v[1]){
            sum+=v[j];
            j=j/2;
        }
        i++;
    }
    return sum;
}
```

Seleccione una:

- (•) a. Las complejidades en el mejor y en el peor de los casos no coinciden.
- ( ) b. El mejor de los casos se da cuando $n = 0$, su complejidad es constante.
- ( ) c. Esta función no presenta casos mejor y peor puesto que sólo puede haber una instancia para cada una de las posibles tallas