

<b>Pregunta 1</b> Incorrecta Puntúa como 1,00 🚩 Marcar pregunta	<p>Un tubo de <math>n</math> centímetros de largo se puede cortar en segmentos de 1 centímetro, 2 centímetros, etc. Existe una lista de los precios a los que se venden los segmentos de cada longitud. Una de las maneras de cortar el tubo es la que más ingresos nos producirá. Di cuál de estas tres afirmaciones es falsa.</p> <p>Seleccione una:</p> <p><input type="radio"/> a. Hacer una evaluación exhaustiva "de fuerza bruta" de todas las posibles maneras de cortar el tubo consume un tiempo <math>\Theta(2^n)</math>. <b>✗</b></p> <p><input type="radio"/> b. Es posible evitar hacer la evaluación exhaustiva "de fuerza bruta" guardando, para cada posible longitud <math>j &lt; n</math> el precio más elevado posible que se puede obtener dividiendo el tubo correspondiente.</p> <p><input type="radio"/> c. Hacer una evaluación exhaustiva "de fuerza bruta" de todas las posibles maneras de cortar el tubo consume un tiempo <math>\Theta(n!)</math>.</p>
<b>Pregunta 2</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>Supongamos que una solución recursiva a un problema de optimización muestra estas dos características: por un lado, se basa en obtener soluciones óptimas a problemas parciales más pequeños, y por otro, estos subproblemas se resuelven más de una vez durante el proceso recursivo. Este problema es candidato a tener una solución alternativa basada en ...</p> <p>Seleccione una:</p> <p><input type="radio"/> a. ... un algoritmo voraz.</p> <p><input checked="" type="radio"/> b. ... un algoritmo de programación dinámica. <b>✓</b></p> <p><input type="radio"/> c. ... un algoritmo del estilo de <i>divide y vencerás</i>.</p>
<b>Pregunta 3</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>¿Cuál de estos tres problemas de optimización no tiene, o no se le conoce, una solución voraz óptima?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. El problema de la mochila continua o con fraccionamiento.</p> <p><input checked="" type="radio"/> b. El problema de la mochila discreta o sin fraccionamiento. <b>✓</b></p> <p><input type="radio"/> c. El árbol de cobertura de coste mínimo de un grafo conexo.</p>
<b>Pregunta 4</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>En el método voraz ...</p> <p>Seleccione una:</p> <p><input checked="" type="radio"/> a. ... es habitual preparar los datos para disminuir el coste temporal de la función que determina cuál es la siguiente decisión a tomar. <b>✓</b></p> <p><input type="radio"/> b. ... siempre se encuentra solución pero puede que no sea la óptima.</p> <p><input type="radio"/> c. ... el dominio de las decisiones sólo pueden ser conjuntos discretos o discretizables.</p>
<b>Pregunta 5</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>La solución de programación dinámica iterativa del problema de la mochila discreta ...</p> <p>Seleccione una:</p> <p><input type="radio"/> a. ... calcula menos veces el valor de la mochila que la correspondiente solución de programación dinámica recursiva.</p> <p><input type="radio"/> b. ... tiene un coste temporal asintótico exponencial con respecto al número de objetos.</p> <p><input checked="" type="radio"/> c. ... tiene la restricción de que los valores tienen que ser enteros positivos. <b>✓</b></p>
<b>Pregunta 6</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>Quando se calculan los coeficientes binomiales usando la recursión <math>\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}</math>, con <math>\binom{n}{0} = \binom{n}{n} = 1</math>, qué problema se da y cómo se puede resolver?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. La recursión puede ser infinita y por tanto es necesario organizarla según el esquema iterativo de programación dinámica.</p> <p><input checked="" type="radio"/> b. Se repiten muchos cálculos y ello se puede evitar usando programación dinámica. <b>✓</b></p> <p><input type="radio"/> c. Se repiten muchos cálculos y ello se puede evitar haciendo uso de una estrategia voraz.</p>
<b>Pregunta 7</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>Dado un problema de optimización, el método voraz ...</p> <p>Seleccione una:</p> <p><input type="radio"/> a. ... siempre obtiene la solución óptima.</p> <p><input checked="" type="radio"/> b. Ninguna de las otras dos opciones es cierta. <b>✓</b></p> <p><input type="radio"/> c. ... siempre obtiene una solución factible.</p>
<b>Pregunta 8</b> Correcta Puntúa como 1,00 🚩 Marcar pregunta	<p>En la solución al problema de la mochila continua ¿por qué es conveniente la ordenación previa de los objetos?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. Porque si no se hace no es posible garantizar que la toma de decisiones siga un criterio voraz.</p> <p><input type="radio"/> b. Para reducir la complejidad temporal en la toma de cada decisión: de <math>O(n^2)</math> a <math>O(n \log n)</math>, donde <math>n</math> es el número de objetos a considerar.</p> <p><input checked="" type="radio"/> c. Para reducir la complejidad temporal en la toma de cada decisión: de <math>O(n)</math> a <math>O(1)</math>, donde <math>n</math> es el número de objetos a considerar. <b>✓</b></p>

<p><b>Pregunta 9</b></p> <p>Correcta</p> <p>Puntúa como 1,00</p> <p>🚩 Marcar pregunta</p>	<p>¿Cuál de los siguientes pares de problemas son equivalentes en cuanto al tipo de solución (óptima, factible, etc.) aportada por el método voraz?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. El fontanero diligente y la asignación de tareas.</p> <p><input type="radio"/> b. El fontanero diligente y el problema del cambio.</p> <p><input checked="" type="radio"/> c. El fontanero diligente y la mochila continua. ✓</p>
<p><b>Pregunta 10</b></p> <p>Correcta</p> <p>Puntúa como 1,00</p> <p>🚩 Marcar pregunta</p>	<p>Se pretende implementar mediante programación dinámica iterativa la función recursiva:</p> <pre>int f( int x, int y ) {     if( x &lt;= y ) return 1;     return x + f(x-1, y); }</pre> <p>¿Cuál es la mejor complejidad espacial que se puede conseguir?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. <math>O(x^2)</math></p> <p><input type="radio"/> b. <math>O(x)</math></p> <p><input checked="" type="radio"/> c. <math>O(1)</math> ✓</p>
<p><b>Pregunta 11</b></p> <p>Sin contestar</p> <p>Puntúa como 1,00</p> <p>🚩 Marcar pregunta</p>	<p>Dada la suma de la recurrencia</p> $T(n) = \begin{cases} 1 & n = 0 \\ \sum_{k=0}^{n-1} T(k) & n > 0 \end{cases}$ <p>¿cuál de las siguientes afirmaciones es cierta?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. <math>T(n) \in \Theta(2^n)</math></p> <p><input type="radio"/> b. <math>T(n) \in \Theta(n^2)</math></p> <p><input type="radio"/> c. <math>T(n) \in \Theta(n!)</math></p>
<p><b>Pregunta 12</b></p> <p>Correcta</p> <p>Puntúa como 1,00</p> <p>🚩 Marcar pregunta</p>	<p>Se pretende implementar mediante programación dinámica recursiva la función recursiva:</p> <pre>float f(unsigned x, int y){     if( y &lt; 0 ) return 0;     float a = 0.0;     if ( v1[y] &lt;= x )         a = v2[y] + f( x-v1[y], y-1 );     float b = f( x, y-1 );     return min(a,2+b); }</pre> <p>¿Cuál es la mejor estructura para el almacén?</p> <p>Seleccione una:</p> <p><input type="radio"/> a. unsigned A[]</p> <p><input checked="" type="radio"/> b. unsigned A[][] ✓</p> <p><input type="radio"/> c. unsigned A</p>