DIVIDE Y VENCERAS Y PROGRAMACIÓN DINAMICA: EJERCICIOS

N	ENUNCIADO
1	¿Qué esquema algorítmico utiliza el algoritmos de ordenación Quicksort? a. Divide y Vencerás b. Programación Dinámica c. Backtracking
2	Ante un problema que presenta una solución recursiva siempre podemos aplicar: a. Divide y vencerás b. Programación dinámica c. Cualquiera de las dos anteriores
3	En cual de los siguientes casos no se puede aplicar el esquema Divide y Vencerás: a. Cuando los subproblemas son de tamaños muy diferentes b. Cuando el problema no cumple el principio de optimalidad c. Se puede aplicar en ambos casos.
4	Dado el algoritmo de búsqueda binaria, supongamos que, en vez de dividir la lista de elementos en dos mitades del mismo tamaño, la dividamos en dos partes de tamaños 1/3 y 2/3. El coste de este algoritmo: a. Es el mismo que el del original b. Es mayor que el del original c. Es menor que el del original
5	Si n es el número de elementos del vector, el coste del algoritmo Mergesort es: a. $O(n^2)$ y $\Omega(n\ logn)$ b. $\Theta\ (n\ logn)$ c. $\Theta\ (n^2)$
6	Un problema se puede resolver por Divide y Vencerás siempre que: a. Cumpla el principio de optimalidad b. Cumpla el teorema de reducción c. Ninguna de las anteriores
7	La serie de números de Fibonacci se define de la siguiente forma: $fib(n) = \begin{cases} 1 & n \le 1 \\ fib(n-1) + fib(n-2) & n > 1 \end{cases}$ Para implementar esta función podemos emplear : a. Divide y vencerás b. Programación dinámica c. Cualquiera de las dos anteriores
8	La serie de números de Fibonacci se define de la siguiente forma: $fib(n) = \begin{cases} 1 & n \le 1 \\ fib(n-1) + fib(n-2) & n > 1 \end{cases}$ ¿Qué implementación de entre las siguientes supone el menor coste? a. Divide y vencerás b. Programación dinámica c. Ambas tienen el mismo coste asintótico
9	El problema de la mochila, ¿puede solucionarse de forma óptima empleando la estrategia de divide y vencerás?: a. Sólo para el caso de la mochila con fraccionamiento b. Sólo para el caso de la mochila sin fraccionamiento c. Si, se puede aplicar para ambos casos.
10	Para que un problema de optimización se pueda resolver mediante PD es necesario que: a. Cumpla el principio de optimalidad b. Cumpla el teorema de reducción c. Cumpla los dos anteriores
11	Dada una solución recursiva a un problema ¿Cómo podemos evitar la resolución de los mismos subproblemas muchas veces? a. Resolver los subproblemas de mayor a menor y guardar su resultado en una tabla, inicializándola con los problemas pequeños. b. Resolver los subproblemas de menor a mayor y guardar su resultado en una tabla, inicializándola con los problemas pequeños. c. Resolver los subproblemas de mayor a menor y guardar su resultado en una tabla, inicializándola con los problemas más grandes.
12	Si aplicamos Programación Dinámica a un problema que también tiene solución por divide y vencerás podemos asegurar que... a. El coste temporal se reduce y el espacial aumenta con respecto a la solución por DyV b. El coste temporal aumenta y el espacial se reduce con respecto a la solución por DyV c. Ninguna de las anteriores.
13	¿Cuándo utilizaremos Programación Dinámica en lugar de Divide y Vencerás? a. Cuando se incrementa la eficacia b. Cuando se incrementa la eficiencia c. Cuando se reduce el coste espacial.
14	En programación dinámica, dónde almacenamos los valores de los problemas resueltos? a. En un vector unidimensional b. En un vector bidimensional c. Depende del problema

15	Supongamos el problema de la mochila resuelto mediante Programación Dinámica y particularizado para n elementos y un peso máximo trasportable de P. ¿Es necesario calcular valores para toda la matriz auxiliar para obtener el resultado? a. Si b. No c. Depende de los valores de n y P.
16	Un problema de optimización cuya solución se puede expresar mediante una secuencia de decisiones cumple el principio de optimalidad si, dada una secuencia óptima: a. Existe una subsecuencia de esa solución que corresponde a la solución óptima de su subproblema asociado b. Existe al menos una subsecuencia de esa solución que corresponde a la solución óptima de su subproblema asociado c. Cualquier subsecuencia de esa solución corresponde a la solución óptima de su subproblema asociado
17	La programación dinámica, para resolver un problema, aplica la estrategia... a. Se resuelven los problemas más pequeños y, combinando las soluciones, se obtienen las soluciones de problemas sucesivamente más grandes hasta llegar al problema original. b. Se descompone el problema a resolver en subproblemas más pequeños, que se resuelven independientemente para finalmente combinar las soluciones de los subproblemas para obtener la solución del problema original. c. Ninguna de las anteriores
18	¿Qué esquema de programación es el adecuado para resolver el problema del k-ésimo mínimo en un vector? a. Programación Dinámica b. Divide y Vencerás c. Ninguno de los dos
19	Si n es el número de elementos de un vector. La solución de menor coste al problema de encontrar su k-ésimo mínimo tiene la siguiente complejidad: a. Ω (n) y O(n logn) b. Ω (n) y O(n^2) c. Ninguna de las dos
20	Si n es el número de elementos de un vector. Podemos encontrar una solución al problema de encontrar su k-ésimo que esté acotada superiormente por : a. O(n^3) b. O (n) c. Ninguna de las dos
21	Dada la solución recursiva al problema de encontrar el k-ésimo mínimo de un vector. Cada llamada recursiva, ¿cuántas nuevas llamadas recursivas genera? a. una o ninguna b. dos o ninguna c. una o dos
22	La solución al problema de encontrar el k-ésimo mínimo de un vector pone en práctica la siguiente estrategia: a. Ordena totalmente el vector b. Ordena parcialmente el vector c. No ordena ningún elemento del vector
23	¿Qué esquema de programación es el adecuado para resolver el problema de la búsqueda binaria? a. Programación Dinámica b. Divide y Vencerás c. Ninguno de los dos
24	Si n es el número de elementos de un vector. La solución de menor coste al problema de la búsqueda binaria tiene la siguiente complejidad: a. Ω (logn) y O(n logn) b. Θ(n logn) c. Ω (1) y O(logn)
25	¿Con qué esquema de programación obtenemos algoritmos que calculan la distancia de edición entre dos cadenas? a. Programación Dinámica b. Divide y vencerás c. Ambos
26	Disponemos de dos cadenas de longitudes m y n. Si resolvemos el problema de la distancia de edición mediante programación dinámica, ¿De qué tamaño debemos definir la matriz que necesitaremos? a. (m-1) x (n-1) b. m x n c. (m+1) x (n+1)