

## **RECOPIACIÓN TESTS EXÁMENES ADA**

1. Dado un problema de optimización cualquiera, ¿la estrategia de vuelta atrás garantiza la solución óptima?

**Es condición necesaria que el dominio de las decisiones sea discreto o discretizable y que el número de decisiones a tomar esté acotado.**

2. En los algoritmo de ramificación y poda...

**Una cota optimista es necesariamente un valor insuperable, de no sea así se podría podar el nodo que conduce a la solución óptima.**

3. La solución recursiva ingenua (pero correcta) a un problema de optimización llama más de una vez a la función con los mismo parámetros. Una de las siguientes afirmaciones es falsa.

**Se puede mejorar la eficiencia del algoritmo convirtiendo el algoritmo recursivo directamente en iterativo sin cambiar su funcionamiento básico.**

4. Si un problema de optimización lo es para una función que toma valores continuos...

**La programación dinámica recursiva puede resultar mucho más eficiente que la programación dinámica iterativa en cuanto al uso de memoria.**

5. El uso de funciones de cota en ramificación y poda...

**... puede reducir el número de instancias del problema que perteneces al caso peor.**

6. Al resolver el problema del viajante de comercio mediante vuelta atrás, ¿cuál de estas cotas optimistas se espera que puede mejor el árbol de búsqueda?

**Se ordenan las aristas restantes de menor a mayor distancia y se calcula la suma de las k aristas más cortas, donde k es el número de saltos que nos quedan por dar.**

7. ¿Para cuál de estos problemas de optimización existe una solución voraz?

**El árbol de recubrimiento mínimo para un grafo no dirigido con pesos.**

8. Se desea encontrar el camino más corto entre dos ciudades. Para ello se dispone de una tabla con la distancia entre los pares de ciudades en los que hay carreteras o un valor centinela (por ejemplo -1) si no hay, por lo que para ir de la ciudad inicial a la final es posible que haya que pasar por varias ciudades. También se conocen las coordenadas geográficas de cada ciudad y por tanto la distancia geométrica (en línea recta) entre cada par de ciudades. Se pretende acelerar la búsqueda de un algoritmo de ramificación y poda priorizando los nodos vivos (ciudades) que estén a menor distancia geográfica de ciudad objetivo.

**El nuevo algoritmo no garantiza que vaya a ser más rápido para todas las instancias del problema posibles.**

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

**... es una función del tamaño o talla del problema que tiene que estar definida para todos los posibles valores de ésta.**

10. La mejor solución que se conoce para el problema de la mochila continua sigue el esquema...

... voraz.

11. La complejidad en el mejor de los casos de un algoritmo de ramificación y poda ...  
**... puede ser polinómica con el número de decisiones a tomar.**

12. Cuando se usa un algoritmo voraz para abordar la resolución de un problema de optimización por selección discreta (es decir, un problema para el cual la solución consiste en encontrar un subconjunto del conjunto de elementos que optimiza una determinada función), ¿cuál de estas tres cosas es imposible que ocurra?

**Que se reconsidere la decisión ya tomada anteriormente respecto a la selección de un elemento a la vista de la decisión que se debe tomar en el instante actual.**

13. Cuando la descomposición recursiva de un problema da lugar a subproblemas de tamaño similar, ¿qué esquema promete ser más apropiado?

**Programación dinámica.**

14. Sea la siguiente relación de recurrencia

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Si  $T(n) \in O(n^2)$ , ¿en cuál de estos tres casos nos podemos encontrar?

**$g(n) = n^2$**

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

**... será más eficiente cuanto más equitativa sea la división en subproblemas.**

16. En el esquema de vuelta atrás, los mecanismos de poda basados en la mejor solución hasta el momento...

**... pueden eliminar soluciones parciales que son factibles.**

17. Uno de estos tres problemas no tiene una solución eficiente que siga el esquema de programación dinámica.

**El problema de las torres de Hanoi.**

18. El valor que se obtiene con el método voraz para el problema de la mochila discreta es...

**... una cota inferior para el valor óptimo que a veces puede ser igual a este.**

19. Decid cuál de estas tres es la cota pesimista más ajustada al valor óptimo de la mochila discreta:

**El valor de la mochila discreta que se obtiene usando un algoritmo voraz basado en el valor específico de los objetos.**

20. Un problema de tamaño  $n$  puede transformarse en tiempo  $O(n^2)$  en otro de tamaño  $n - 1$ . Por otro lado, la solución al problema cuando la talla es 1 requiere un tiempo constante.

¿Cuál de estas clases de coste temporal es la más ajustada?

**$O(n^3)$**

21. El siguiente programa resuelve el problema de cortar un tubo de longitud  $n$  en segmentos de longitud entera entre 1 y  $n$  de manera que se maximice el precio de acuerdo con una tabla que da el precio para cada longitud, pero falta un trozo. ¿Que debería ir en lugar de XXXXXXXX?

```
void fill(price r[]) {
    for (index i=0;i<=n;i++) r[i]=-1;
}

price cutrod(price p[], r[], length n) {
    price q;
    if (r[n]>=0) return r[n];
    if (n==0) q=0;
    else {
        q=-1;
        for (index i=1;i<=n;i++)
            q=max(q,p[i]+cutrod(XXXXXXX));
    }
    r[n]=q;
    return q;
}
```

- a. p, r-1, n
- b. p, n-r[n]
- \*c. p, r, n-i

22. Una de estas tres situaciones no es posible:  
 $f(n) \in \Omega(n^2)$  y  $f(n) \in O(n)$

23. Sea  $A$  una matriz cuadrada  $n \times n$ . Se trata de buscar una permutación de las columnas tal que la suma de los elementos de la diagonal de la matriz resultante se mínima. Indica cuál de las siguientes afirmaciones es falsa.  
**La complejidad temporal de la mejor solución posible al problemas es  $O(n^2)$ .**

24. Cuál de los siguientes algoritmos proveería una cota pesimista para el problema de encontrar el camino más corto entre dos ciudades (se suponen que el grafo es conexo)  
**Calcular la distancia recorrida moviéndose al azar por el grafo hasta llegar (por azar) a la ciudad destino.**

25. Si para resolver un mismo problema usamos un algoritmo de vuelta atrás y lo modificamos mínimamente para convertirlo en un algoritmo de ramificación y poda, ¿qué cambiamos realmente?  
**El algoritmo puede aprovechar mejor las cotas optimistas.**

26. Dadas las siguientes funciones:

```
// Precondición: { 0 <= i < v.size(); i < j <= v.size() }
unsigned f( const vector<unsigned>&v, unsigned i, unsigned j ) {
    if( i == j+1 )
        return v[i];
    unsigned sum = 0;
    for( unsigned k = 0; k < j - i; k++ )
        sum += f( v, i, i+k+1 ) + f( v, i+k+1, j );
    return sum;
}

unsigned g( const vector<unsigned>&v ) {
    return f( v, v.begin(), v.end() );
}
```

Se quiere reducir la complejidad temporal de la función g usando programación dinámica iterativa, ¿cuál sería la complejidad espacial?

**cuadrática**

27. En una cuadrícula se quiere dibujar el contorno de un cuadrado de n casillas de lado. ¿cuál será la complejidad temporal del mejor algoritmo que pueda existir?

**O(n)**

28. En los algoritmos de ramificación y poda, ¿el valor de una cota pesimista es mayor que el valor de una cota optimistas? (se entiende que ambas cotas se aplican sobre el mismo nodo)

**En general sí, si se trata de un problema de minimización, aunque en ocasiones ambos valores pueden coincidir.**

29. Cuando se resuelve el problema de la mochila discreta usando la estrategia de vuelta atrás, ¿puede ocurrir que se tarde menos en encontrar la solución óptima si se prueba primero a meter cada objeto antes de no meterlo?

**Sí, pero sólo si se usan cotas optimistas para podar el árbol de búsqueda.**

30. Garantiza el uso de una estrategia “divide y vencerás” la existencia de una solución de complejidad temporal polinómica a cualquier problema?

**No**

31. En el esquema vuelta atrás el orden en el que se van asignando los distintos valores a las componentes del vector que contendrá la solución...

**Las otras dos opciones son ciertas.**

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

**... se comporta peor cuando el vector ya está ordenado.**

33. ¿Cual de estos tres problemas de optimización no tiene, o no se le conoce, una solución voraz (greedy) que se óptima?

**El problema de la mochila discreta.**

34. En un problema de optimización, si el dominio de las decisiones es un conjunto infinito,

**una estrategia voraz puede ser la única alternativa**

35. Dado un problema de optimización, el método voraz...

**garantiza la solución óptima sólo para determinados problemas.**

36. La mejora que en general aporta la programación dinámica frente a la solución ingenua se consigue gracias al hecho de que...

**... en la solución ingenua se resuelve muchas veces un número relativamente pequeño de subproblemas distintos.**

37. Se quieren ordenar  $d$  números distintos comprendidos entre 1 y  $n$ . Para ellos se usa un array de  $n$  booleanos que se inicializan primero a false. A continuación se recorren los  $d$  números cambiando los valores del elemento del vector de booleanos correspondiente a su número true. Por último se recorre el vector de booleanos escribiendo los índices de los elementos del vector de booleanos que son true. ¿Es este algoritmo más rápido (asintóticamente) que el mergesort?

**Sólo si  $d \log d > k n$  (donde  $k$  es una constante que depende de la implementación).**

38. ¿Cual de estos problemas tiene una solución eficiente utilizando programación dinámica?

**El problema del cambio**

39. Di cuál de estos tres algoritmos no es un algoritmo “divide y vencerás”

**El algoritmo de Prim**

40. Si  $f(n) \in O(n^3)$ , ¿puede pasar que  $f(n) \in O(n^2)$ ?

**Es perfectamente posible, ya que  $O(n^2) \subset O(n^3)$**

41. Tratándose de un esquema general para resolver problemas de minimización, ¿qué falta en el hueco?

```
Solution BB( Problem p ) {
    Node best, init = initialNode(p);
    Value pb = init.pessimistic_b();
    priority_queue<Node>q.push(init);
    while( ! q.empty() ) {
        Node n = q.top(); q.pop();
        q.pop();
        if( ????????? ) {
            pb = max( pb, n.pessimistic_b());
            if( n.isTerminal() )
                best = n.sol();
            else
                for( Node n : n.expand() )
                    if( n.isFeasible() )
                        q.push(n);
        }
    }
    return best;
}
```

**$n.optimistic\_b() \leq pb$**

42. ¿Cuál de estas estrategias para calcular el n-ésimo elemento de la serie de Fibonacci ( $f(n) = f(n-1) + f(n-2)$ ,  $f(1) = f(2) = 1$ ) es más eficiente?

**Programación dinámica.**

43. En los algoritmos de ramificación y poda, ¿el valor de una cota pesimista es menor que el valor de una cota optimista? (se entiende que ambas cotas se aplican sobre el mismo nodo)

**En general sí, si se trata de un problema de maximización, aunque en ocasiones ambos valores pueden coincidir.**

44. En un algoritmo de ramificación y poda, el orden escogido para priorizar los nodos en la lista de nodos vivos...

**... pueden influir en el número de nodos que se descartan sin llegar a expandirlos.**

45. El algoritmo de ordenación Quicksort divide el programa en dos subproblemas ¿Cuál es la complejidad temporal asintótica de realizar esa división?

**$O(n)$**

46. Uno de estos tres problemas no tiene una solución trivial y eficiente que siga el esquema voraz.

**El problema del cambio.**

47. Estudiar la relación de recurrencia:

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ pT(\frac{n}{q}) + g(n) & \text{en otro caso} \end{cases}$$

(donde p y q son enteros mayores que 1). Di cuál de los siguientes esquemas algorítmicos produce de manera natural relaciones de recurrencia así.

**Divide y vencerás**

48. El siguiente programa resuelve el problema de cortar un tubo de longitud n en segmentos de longitud entera entre 1 y n de manera que se maximice el precio de acuerdo con una tabla que da el precio para cada longitud, pero falta un trozo. ¿Que debería ir en lugar de XXXXXXXX?

```
void fill(price m[]) {
    for (index i=0; i<=n; i++) m[i]=-1;
}

price cutrod(length n, price m[], price p[]) {
    price q;
    if (m[n]>=0) return m[n];
    if (n==0) q=0;
    else {
        q=-1;
        for (index i=1; i<=n; i++)
            q=max(q, p[i]+cutrod(XXXXXXX));
    }
    m[n]=q;
    return q;
}
```

**n-i, m, p**

49. Sea  $g(n) = \sum a_i \cdot n^i$  (hasta K con  $i = 0$ ). Di cuál de las siguientes afirmaciones es falsa:  
**Las otras dos afirmaciones son ambas falsas.**

50. Si  $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$  entonces...  
**...  $f(n) \in O(g(n))$**

51. ¿Cuál es la diferencia principal entre una solución de vuelta atrás y una solución de ramificación y poda para el problema de la mochila?

**El orden de exploración de las soluciones.**

52. En un algoritmo de ramificación y poda, si la lista de nodos vivos no está ordenada de forma apropiada...

**... podría ocurrir que se exploren nodos de forma innecesaria.**

53. Sea la siguiente relación de recurrencia

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Si  $T(n) \in O(n)$ , ¿en cuál de estos tres casos nos podemos encontrar?  
 **$g(n) = 1$**

54. Los algoritmos de vuelta atrás que hacen uso de cotas optimistas generan las soluciones posibles al problema mediante ...

**... un recorrido en profundidad del árbol que representa el espacio de soluciones.**

55. ¿Qué tienen en común el algoritmo que obtiene el k-ésimo elemento más pequeño de un vector (estudiado en clase) y el algoritmo de ordenación Quicksort?

**La división del problema en subproblemas.**

56. ¿Cuál es el coste temporal asintótico de la siguiente función?

```
void f(int n, int arr[]) {  
    int i = 0, j = 0;  
    for(; i < n; ++i)  
        while(j < n && arr[i] < arr[j])  
            j++;  
}
```

**$O(n)$**

57. ¿Se puede reducir el coste temporal de un algoritmo recursivo almacenando los resultados devueltos por las llamadas recursivas?

**Sí, si se repiten llamadas a la función con los mismos argumentos.**

58. Sea  $n$  el número de elementos que contienen los vectores  $w$  y  $v$  en la siguiente función  $f$ . ¿Cuál es su complejidad temporal asintótica en función de  $n$  asumiendo que en la llamada inicial el parámetro  $i$  toma valor  $n$ ?

```
float f(vector<float>&w, vector<unsigned>&v,
        unsigned P, int i){
    float S1, S2;
    if (i>=0){
        if (w[i] <= P)
            S1= v[i] + f(w,v,P-w[i],i-1);
        else S1= 0;
        S2= f(w,v,P,i-1);
        return max(S1,S2);
    }
    return 0;
}
```

$\Omega(n)$  y  $O(2^n)$

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

**... se comporta mejor cuando el vector ya está ordenado.**

60.Cuál de los siguientes criterios proveería una cota optimista para el problema de encontrar el camino más corto entre dos ciudades (se supone que el grafo es conexo).

**Calcular la distancia geométrica (en línea recta) entre la ciudad origen y destino.**

61. La siguiente relación de recurrencia expresa la complejidad de un algoritmo recursivo, donde  $g(n)$  es una función polinómica:

$$T(n) = \begin{cases} 1 & \text{si } n \leq 1 \\ 2T(\frac{n}{2}) + g(n) & \text{en otro caso} \end{cases}$$

Di cuál de las siguientes afirmaciones es falsa:

**Si  $g(n) \in O(n^2)$  la relación de recurrencia representa la complejidad temporal del algoritmo de búsqueda por inserción.**

62. ¿Cuál es la definición correcta de  $O(f)$ ?

.... (PONER A MANO)

63. Si  $f(n) \in O(n^2)$ , ¿podemos decir siempre que  $f(n) \in O(n^3)$ ?

**Si, ya que  $n^2 \in O(n^3)$**

64. Se desea encontrar el camino más corto entre dos ciudades. Para ello se dispone de una tabla con la distancia entre los pares de ciudades en los que hay carreteras o un valor centinela (por ejemplo, -1) si no hay, por lo que para ir de la ciudad inicial al a final es posible que haya que pasar por varias ciudades. Como también se conocen las



coordenadas geográficas de cada ciudad se quiere usar la distancia geográfica (en línea recta) entre cada par de ciudades como cota para limitar la búsqueda en un algoritmo de vuelta atrás. ¿Que tipo de cota sería?

**Una cota optimista.**

65. Sea A una matriz cuadrada  $n \times n$ . Se trata de buscar una permutación de las columnas tal que la suma de los elementos de la diagonal de la matriz resultante sea mínima. Indicad cuál de las siguientes afirmaciones es falsa.

**La complejidad temporal de la mejor solución posible al problema es  $O(n \log n)$**

66. La complejidad temporal de la solución de vuelta atrás al problema de la mochila discreta es...

**... exponencial en el caso peor.**

67. Ante un problema de optimización resuelto mediante backtracking, ¿Puede ocurrir que el uso de las cotas pesimistas y optimistas sea inútil incluso perjudicial?

**Sí, puesto que es posible que a pesar de utilizar dichas cotas no se descarte ningún nodo.**

68. ¿Qué se entiende por tamaño del problema?

**La cantidad de espacio en memoria que se necesita para codificar una instancia de ese problema.**

69. Si  $\lim_{n \rightarrow \infty} (f(n)/n^2) = k$ , y  $k \neq 0$ , ¿cuál de estas tres afirmaciones es falsa?

**$f(n) \in \Theta(n^3)$**

70. ¿Cuál es la definición correcta de  $\Omega(g)$ ?

**... PONER A MANO**

71. ¿Cuál es la definición correcta de  $\Omega(f)$ ?

**... PONER A MANO**

72. Tratándose de un esquema general para resolver problemas de maximización, ¿qué falta en el hueco?.

```

Solution BB( Problem p ) {
Node best, init = initialNode(p);
Value pb = init.pessimistic_b();
priority_queue<Node>q;
q.push(init);
while( ! q.empty() ) {
    Node n = q.top(); q.pop();
    q.pop();
    if( ????????? ) {
        pb = max( pb, n.pessimistic_b());
        if( n.isTerminal() )
            best = n.sol();
        else
            for( Node n : n.expand() )
                if( n.isFeasible() )
                    q.push(n);
    }
}
return best;
}

```

**n.optimistic\_b() >= pb**

73. De las siguientes expresiones, o bien dos verdades y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta a las otras dos.

**$\Theta(\log^2(n)) = \Theta(\log^3(n))$**

74. La función  $y$  de un número semientero positivo (un número es semientero si al restarle 0.5 es entero) se define como:

```
double gamma( double n ) { // Se asume  $n \geq 0.5$  y  $n-0.5$  entero
    if( n == 0.5 )
        return sqrt(PI);
    return n * gamma( n - 1 );
}
```

¿Se puede calcular usando programación dinámica iterativa?

**Si, pero la complejidad temporal no mejora.**

75. ¿Cuál de estas afirmaciones es falsa?

**Los algoritmos iterativos de programación dinámica utilizan memorización para evitar resolver de nuevo los mismos subproblemas que se vuelve a presentar**

76. El algoritmo de ordenación Mergesort divide el problema en dos subproblemas. ¿Cuál es la complejidad temporal asintótica de realizar esa división?

**$O(1)$**

77. Supongamos el algoritmo de ordenación Mergesort modificado de manera que, en lugar de dividir el vector en dos partes, se divide en tres. Posteriormente se combinan las soluciones parciales. ¿Cuál sería la complejidad temporal del nuevo algoritmo?

**$n \log(n)$**

78. Los algoritmos voraces...

**... se basan en la idea de que la solución óptima se puede construir añadiendo repetidamente el mejor elemento disponible.**

79. En los algoritmos de backtracking, ¿Puede el valor de una cota pesimista ser mayor que el valor de una cota optimista? (se entiende que ambas cota se aplican sobre el mismo nodo)

**En general sí, si se trata de un problema de minimización, aunque en ocasiones ambos valores pueden coincidir.**

80. ¿Cuál de estas afirmaciones es falsa?

**Todos los problemas de optimización tienen una solución voraz que es óptima sea cual se la instancia a resolver.**

81. De las siguientes expresiones, o bien dos son verdaderas y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta de las otras dos.

**$O(n^2)$  C  $O(2^{\log_2(n)})$  C  $O(2^n)$**

82. Si  $f \in \Theta(g_1)$  y  $f \in \Theta(g_2)$  entonces

**Las otras dos opciones son ambas ciertas**

83. La siguiente relación de recurrencia expresa la complejidad de un algoritmo recursivo, donde  $g(n)$  es una función polinómica:

(FOTO PAG 18 EJE 34)

Di cuál de las siguientes afirmaciones es falsa:

**Si  $g(n) \in \Theta(1)$  la relación de recurrencia representa la complejidad temporal del algoritmo de búsqueda dicotómica.**

84. ¿Para qué puede servir la cota pesimista de un nodo de ramificación y poda?

**Para actualizar el valor de la mejor solución hasta el momento.**

85. Dada la relación de recurrencia:

FOTO PAG 19 EJE 39)

(donde  $p$  y  $a$  son enteros mayores que 1 y  $g(n) = n^k$ ), ¿qué tiene que ocurrir para que se cumpla  $T(n) \in \Theta(n^k \log_a(n))$ ?

**$p = a^k$**

86. Sea  $A$  una matriz cuadrada  $n \times n$ . Se trata de buscar una permutación de las columnas tal que la suma de los elementos de la diagonal de la matriz resultante sea mínima. Indica cuál de las siguientes afirmaciones es correcta.

**Si se construye una solución al problema basada en el esquema de ramificación y poda, una buena elección de cotas optimistas y pesimistas podría evitar la exploración de todas las permutaciones posibles.**

87. ¿Cuál es la complejidad temporal en el mejor de los casos de la siguiente función?

(PAG 19 EJE7 FOTO)

**a.  $\Omega(n)$**

88. En el problema del coloreado de grafos (mínimo número de colores necesarios para colorear todos los vértices de un grafo de manera que no queden dos adyacentes con el mismo color) resuelto mediante ramificación y poda, una cota optimista es el resultado de asumir que...

**... no se van a utilizar colores distintos a los ya utilizados.**

89. De las siguientes expresiones, o bien dos son verdaderas y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta de las otras dos.  
 **$\Theta(\log_2(n)) = \Theta(\log_3(n))$**

90. Se quiere reducir la complejidad temporal de la siguiente función haciendo uso de programación dinámica. ¿Cuál sería la complejidad temporal resultante?  
(pag 20 eje 13)

**Cuadrática**

91. De las siguientes expresiones, o bien dos son verdaderas y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta de las otras dos.  
 **$O(2^{\log_2(n)})$  C  $O(n^2)$   $O(n!)$**

92. Dada la relación de recurrencia:  
FOTO PAG 20 EJE 15)

(donde p y a son enteros mayores que 1 y  $g(n) = n^k$ ), ¿qué tiene que ocurrir para que se cumpla  $T(n) \in \Theta(n^k)$ ?  
 **$p < a^k$**

93. Si  $\lim_{n \rightarrow \infty} (f(n)/n^2) = k$ , y  $k \neq 0$ , ¿cuál de estas tres afirmaciones es cierta?  
 **$f(n) \in \Theta(n^2)$**

94. Si  $f \in \Theta(g_1)$  y  $f \in \Theta(g_2)$  entonces  
 **$f \in \Theta(g_1 + g_2)$**

95. Supongamos el algoritmo de ordenación Mergesort modificado de manera que, en lugar de dividir el vector en dos partes, se divide en tres. Posteriormente se combinan las soluciones parciales. ¿Cuál sería la complejidad temporal asintótica de la combinación de las soluciones parciales?  
 **$\Theta(n)$**

96. ¿Cuál es la complejidad temporal de la siguiente función?  
(pag 21 eje 25)  
 **$\Theta(\log^2 n)$**

97. De las siguientes expresiones, o bien dos son verdaderas y una es falsa, o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta a las otras dos.  
(pag 21 eje 26 rellena)

98. La siguiente relación de recurrencia expresa la complejidad de un algoritmo recursivo, donde  $g(n)$  es una función polinómica:  
(pag 22 eje 30)

Di cuál de las siguientes afirmaciones es cierta:

**Si  $g(n) \in \Theta(n)$  la relación de recurrencia representa la complejidad temporal en el caso mejor del algoritmo de ordenación quicksort**

99. ¿Cuál de estas afirmaciones es cierta?

**La memorización evita que un algoritmo recursivo ingenuo resuelva repetidamente el mismo problema.**

100. Dado el problema del laberinto con tres movimientos, ¿se puede aplicar un esquema de programación dinámica para obtener un camino de salida?

**Si, en caso de existir con este esquema siempre se puede encontrar un camino de salida.**

101. En ausencia de cotas optimistas y pesimistas, la estrategia de vuelta atrás...

**... no recorre todo el árbol si hay manera de descartar subárboles que representan conjuntos de soluciones no factibles.**

102. Decid cuál de estas tres es la cota optimista más ajustada al valor óptimo de la mochila discreta:

**El valor de la mochila continua correspondiente.**

103. El coste temporal asintótico de insertar un elemento en un vector ordenado de forma que continúe ordenado es...

**...  $O(n)$**

104. ¿En ramificación y poda, tiene sentido utilizar la cota optimista de los nodos como criterio para ordenar la lista de nodos vivos?

**Sí, aunque no es una garantía de que sea una buena estrategia de búsqueda.**

105. ¿Qué estrategia de búsqueda es a priori más apropiada en un esquema de vuelta atrás?

**En el esquema de vuelta atrás no se pueden definir estrategias de búsqueda.**

106. Sea la relación de recurrencia  
(pag 23 eje 7)

Si  $T(n) \in O(n)$ , ¿en cuál de estos tres casos nos podemos encontrar?

**Las otras dos opciones son ambas ciertas.**

107. Si  $f \in \Omega(g_1)$  y  $f \in \Omega(g_2)$  entonces  
 **$f \in \Omega(g_1 + g_2)$**

108. Dada la siguiente función:  
(pag 24 eje 9)

¿Cuál es su complejidad temporal asintótica?  
 **$O(n)$**

109. El esquema de vuelta atrás...

**Garantiza que encuentra la solución óptima a cualquier problema de selección discreta.**

110. En el esquema de ramificación y poda, ¿qué estructura es la más adecuada si queremos realizar una exploración por niveles?

**Cola**

111. ¿Qué nos proporciona la media entre el coste temporal asintótico (o complejidad temporal) en el peor caso y el coste temporal asintótico en el mejor caso?

**Nada de interés**

112. ¿Qué complejidad se obtiene a partir de la relación de recurrencia  $T(n) = 8T(n/2) + n^3$  con  $T(1) = O(1)$ ?

**$O(n^3 \log n)$**

113. Dado el problema del laberinto con tres movimientos, se desea saber el número de caminos distintos desde la casilla inicial (1, 1) hasta la casilla (n, m) y para ello se aplica un esquema de programación dinámica. En cuanto a la complejidad temporal, ¿cuál es la mejora de la versión recursiva con memoización frente a la recursiva ingenua que se obtiene a partir del esquema divide y vencerás?

**De una complejidad exponencial que se obtendría con la ingenua se reduciría a polinómica con la de memoización**

114. Dada la siguiente función:

(pag 25 eje 15)

Marcad la opción correcta.

**La complejidad temporal en el mejor de los casos es  $\Omega(n)$**

115. El esquema voraz...

**Las otras dos opciones son ambas falsas.**

116. Cuando la descomposición de un problema da lugar a subproblemas de tamaño similar al original, muchos de los cuales se repiten, ¿qué esquema es a priori más apropiado?

**Programación dinámica.**

117. Dada la siguiente función (donde  $\max(a, b) \in \Theta(1)$ )

(pag 26 eje 18)



Marcad la opción correcta.

**La complejidad temporal en el peor de los casos es  $O(2^n)$**

118. ¿Cuál sería la complejidad temporal de la siguiente función tras aplicar programación dinámica?

(pag 26 eje 19)

**$\Theta(n)$**

119. Dado el problema del laberinto con tres movimientos, se pretende conocer la longitud del camino de salida más corto. Para ello se aplica el esquema voraz con un criterio de selección que consiste en elegir primero el movimiento Este siempre que la casilla sea accesible. Si no lo es se descarta ese movimiento y se prueba con Sureste y por último, si este tampoco es posible, se escoge el movimiento Sur. ¿Qué se puede decir del algoritmo obtenido?

**Que en realidad no es un algoritmo voraz pues el criterio de selección no lo es.**

120. Si el coste temporal de un algoritmo es  $T(n)$ , ¿cuál de las siguientes situaciones es imposible?

(pag 27 eje 21) escribir a mano

121. Se desea resolver el problema de la potencia enésima ( $x^n$ ), asumiendo que  $n$  es par y que se utilizará la siguiente recurrencia:  $\text{pot}(x, n) = \text{pot}(x, n/2) * \text{pot}(x, n/2)$ ; ¿Qué esquema resultará ser más eficiente en cuanto al coste temporal?

**Programación dinámica**

122. De las siguientes afirmaciones marca la que es verdadera.

**En un esquema de vuelta atrás, las cotas pesimistas no tienen sentido si lo que se pretende es obtener todas las soluciones factibles.**

123. Dado un problema de minimización resuelto mediante un esquema de ramificación y poda, ¿qué propiedad cumple una cota optimista?

**Las otras dos opciones son ambas falsas**

124. Dado el problema del laberinto con tres movimientos, se desea saber el número de caminos distintos desde la casilla inicial (1, 1) hasta la casilla (n, m) y para ello se aplica el esquema de programación dinámica para obtener un algoritmo lo más eficiente posible en cuanto a complejidad temporal y espacial. ¿Cuáles serían ambas complejidades?

(pag 27 eje 25)

125. ¿Qué se deduce de  $f(n)$  y  $g(n)$  si se cumple  $\lim_{n \rightarrow \infty} (f(n)/g(n)) = k$ , con  $k \neq 0$ ?

(pag 27 eje 26)

126. Se desea obtener todas las permutaciones de una lista compuesta por n elementos. ¿Qué esquema es el más adecuado?

**Vuelta atrás, es el esquema más eficiente para este problema.**

127. Dado el problema del laberinto con tres movimientos, se desea saber el número de caminos distintos desde la casilla inicial (1, 1) hasta la casilla (n, m) y para ello se aplica un esquema de divide y vencerás. ¿Cuál sería la recurrencia apropiada para el caso general?

$$nc(n, m) = nc(n - 1, m) + nc(n, m - 1) + nc(n - 1, m - 1)$$

128. Un tubo de n 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. Se quiere resolver el problema mediante vuelta atrás, ¿cuál sería la forma más adecuada de representar las posibles soluciones ?

**un vector de booleanos**

129. De las siguientes expresiones, o bien dos son verdaderas y una es falsa o bien dos son falsas y una es verdadera. Marca la que (en este sentido) es distinta de las otras dos.

(pag 28 eje 30)

130. En el esquema de vuelta atrás, los mecanismos de poda basados en la mejor solución hasta el momento...

**... pueden eliminar vectores que representan posibles soluciones factibles.**

131. En el problema del viajante de comercio (travelling salesmas problem) queremos listar todas las soluciones factibles.

**El orden en el que se exploran las soluciones parciales no es relevante, por ello, la técnica de ramificación y poda no aporta nada con respecto a vuelta atrás.**

132. Dado un problema de maximización resuelto mediante un esquema de ramificación y poda, ¿que ocurre si la cota optimista resulta ser un valor excesivamente elevado?

**Que se podría explorar más nodos de los necesarios**

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

**... alcanza su máxima eficiencia cuando el problema de tamaño  $n$  se divide en  $a$  problemas de tamaño  $n/a$**

134. Dado el problema de las torres de Hanoi resuelto mediante divide y vencerás, ¿cuál de las siguientes relaciones recurrencia expresa mejor su complejidad temporal para el caso general, siendo  $n$  el número de discos?

**$T(n) = 2T(n-1) + 1$**

135. Una de las prácticas de laboratorio consistió en el cálculo empírico de la complejidad temporal promedio del algoritmo de ordenación de vectores Quicksort tomando como centinela el elemento del vector que ocupa la posición central. ¿Cual es el orden de complejidad que se obtuvo?

**$n \log n$**

136. Se desea ordenar una lista enlazada de  $n$  elementos haciendo uso del algoritmo mergesort. En este caso, al tratarse de una lista, la complejidad temporal asintótica de realizar la división en subproblemas resulta ser lineal con el tamaño de esa lista. ¿Cuál sería entonces el coste temporal de realizar dicha ordenación?

a.  $\Theta(n \log n)$

b.  $\Theta(n^2)$

**c. Ninguna de las otras dos opciones es cierta.**

137. (PAG 30 EJE38)

138. Dado el problema del laberinto con tres movimientos, ¿cuál de las estrategias siguientes proveería de una cota optimista para ramificación y poda?

**Las otras dos estrategias son ambas válidas**

139. ¿Qué ocurre si la cota pesimista de un nodo se corresponde con una solución que no es factible?

**Que el algoritmo sería incorrecto pues podría descartarse un nodo que conduce a la solución óptima.**