

Pregunta 1
Sin contestar
Puntúa como 1,00
🚩 Marcar pregunta

La programación dinámica...

Seleccione una:

- ☐ a. ... en algunos casos se puede utilizar para resolver problemas de optimización con dominios continuos pero probablemente pierda su eficacia ya que puede disminuir drásticamente el número de subproblemas repetidos.
- ☒ b. Las otras dos opciones son ciertas. ✓
- ☐ c. ... normalmente se usa para resolver problemas de optimización con dominios discretizables puesto que las tablas se han de indexar con este tipo de valores.

Pregunta 2
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

Si ante un problema de decisión existe un criterio de selección voraz entonces ...

Seleccione una:

- ☐ a. ... la solución óptima está garantizada.
- ☐ b. ... al menos una solución factible está garantizada.
- ☒ c. Ninguna de las otras dos opciones es cierta. ✓

Pregunta 3
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

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

Seleccione una:

- ☒ a. Programación dinámica. ✓
- ☐ b. Divide y vencerás, siempre que se garantice que los subproblemas no son del mismo tamaño.
- ☐ c. El método voraz.

Pregunta 4
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

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

Seleccione una:

- ☐ a. ... una cota superior para el valor óptimo.
- ☒ b. ... una cota inferior para el valor óptimo que a veces puede ser igual a este. ✓
- ☐ c. ... una cota inferior para el valor óptimo, pero que nunca coincide con este.

Pregunta 5
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

Un informático quiere subir a una montaña y para ello decide que tras cada paso, el siguiente debe tomarlo en la dirección de máxima pendiente hacia arriba. Además, entenderá que ha alcanzado la cima cuando llegue a un punto en el que no haya ninguna dirección que sea cuesta arriba. ¿qué tipo de algoritmo está usando nuestro informático?

Seleccione una:

- ☒ a. un algoritmo voraz. ✓
- ☐ b. un algoritmo de programación dinámica.
- ☐ c. un algoritmo divide y vencerás.

Pregunta 6
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

¿Cuál de los siguientes pares de problemas son equivalentes en cuanto al tipo de solución (óptima, factible, etc.) aportada por el método voraz?

Seleccione una:

- ☐ a. El fontanero diligente y el problema del cambio.
- ☐ b. La mochila continua y la asignación de tareas.
- ☒ c. La mochila discreta y la asignación de tareas. ✓

Pregunta 7
Correcta
Puntúa como 1,00
🚩 Marcar pregunta

¿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?

Seleccione una:

- ☐ a. La estrategia voraz.
- ☐ b. Las dos estrategias citadas serían similares en cuanto a eficiencia.
- ☒ c. Programación dinámica. ✓

Pregunta 8

Correcta

Puntúa como 1,00

 Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
int f( int x, int y ) {  
    if( x <= y ) return 1;  
    return x + f(x-1,y);  
}
```

¿Cuál es la mejor complejidad espacial que se puede conseguir?

Seleccione una:

- ☒ a. $O(1)$ ✓
- ☐ b. $O(x^2)$
- ☐ c. $O(x)$

Pregunta 9

Correcta

Puntúa como 1,00

 Marcar pregunta

Dada la suma de la recurrencia

$$T(n) = \begin{cases} 1 & n = 0 \\ \sum_{k=0}^{n-1} T(k) & n > 0 \end{cases}$$

¿cuál de las siguientes afirmaciones es cierta?

Seleccione una:

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

Pregunta 10

Correcta

Puntúa como 1,00

 Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
unsigned f( unsigned x, unsigned v[] ) {  
    if (x==0)  
        return 0;  
    unsigned m = 0;  
    for ( unsigned k = 0; k < x; k++ )  
        m = max( m, v[k] + f( x-k, v ) );  
    return m;  
}
```

¿Cuál es la mejor estructura para el almacén?

Seleccione una:

- ☒ a. int A[] ✓
- ☐ b. int A
- ☐ c. int A[][]

Pregunta 11

Correcta

Puntúa como 1,00

 Marcar pregunta

De los problemas siguientes, indicad cuál no se puede tratar eficientemente como los otros dos:

Seleccione una:

- ☐ a. El problema de cortar un tubo de forma que se obtenga el máximo beneficio posible.
- ☐ b. El problema del cambio, o sea, el de encontrar la manera de entregar una cantidad de dinero usando el mínimo de monedas posibles.
- ☒ c. El problema de la mochila sin fraccionamiento y sin restricciones en cuanto al dominio de los pesos de los objetos y de sus valores. ✓

Pregunta 12

Correcta

Puntúa como 1,00

 Marcar pregunta

La eficiencia de los algoritmos voraces se basa en el hecho de que ...

Seleccione una:

- ☒ a. ... las decisiones tomadas nunca se reconsideran. ✓
- ☐ b. ... antes de tomar una decisión se comprueba si satisface las restricciones del problema.
- ☐ c. ... con antelación, las posibles decisiones se ordenan de mejor a peor.

Pregunta 2

Correcta

Puntúa como 1,00

Marcar pregunta

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

Seleccione una:

- ☐ a. ... en la solución ingenua se resuelve pocas veces un número relativamente grande de subproblemas distintos.
- ☒ b. ... en la solución ingenua se resuelve muchas veces un número relativamente pequeño de subproblemas distintos. ✓
- ☐ c. El número de veces que se resuelven los subproblemas no tiene nada que ver con la eficiencia de los problemas resueltos mediante programación dinámica.

Pregunta 3

Correcta

Puntúa como 1,00

Marcar pregunta

En la solución al problema de la mochila continua ¿por qué es conveniente la ordenación previa de los objetos?

Seleccione una:

- ☒ a. Para reducir la complejidad temporal en la toma de cada decisión: de $O(n)$ a $O(1)$, donde n es el número de objetos a considerar. ✓
- ☐ b. Porque si no se hace no es posible garantizar que la toma de decisiones siga un criterio voraz.
- ☐ c. Para reducir la complejidad temporal en la toma de cada decisión: de $O(n^2)$ a $O(n \log n)$, donde n es el número de objetos a considerar.

Pregunta 6

Sin contestar

Puntúa como 1,00

Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
float f(unsigned x, int y){  
    if( y < 0 ) return 0;  
    float A = 0.0;  
    if ( v1[y] <= x )  
        A = v2[y] + f( x-v1[y], y-1 );  
    float B = f( x, y-1 );  
    return min(A,2+B);  
}
```

¿Cuál es la mejor complejidad espacial que se puede conseguir?

Seleccione una:

- ☐ a. $O(y)$
- ☐ b. $O(1)$
- ☐ c. $O(y^2)$

Pregunta 7

Correcta

Puntúa como 1,00

Marcar pregunta

En el método voraz ...

Seleccione una:

- ☐ a. ... siempre se encuentra solución pero puede que no sea la óptima.
- ☒ b. ... es habitual preparar los datos para disminuir el coste temporal de la función que determina cuál es la siguiente decisión a tomar. ✓
- ☐ c. ... el dominio de las decisiones sólo pueden ser conjuntos discretos o discretizables.

Pregunta 8

Correcta

Puntúa como 1,00

Marcar pregunta

¿ Cómo se vería afectada la solución voraz al problema de la asignación de tareas en el caso de que se incorporaran restricciones que contemplen que ciertas tareas no pueden ser adjudicadas a ciertos trabajadores ?

Seleccione una:

- ☐ a. Ya no se garantizaría la solución óptima pero si una factible.
- ☐ b. Habría que replantearse el criterio de selección para comenzar por aquellos trabajadores con más restricciones en cuanto a las tareas que no pueden realizar para asegurar, al menos, una solución factible.
- ☒ c. La solución factible ya no estaría garantizada, es decir, pudiera ser que el algoritmo no llegue a solución alguna. ✓

Pregunta 9

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
unsigned f( unsigned x, unsigned v[] ) {  
    if (x==0)  
        return 0;  
    unsigned m = 0;  
    for ( unsigned k = 0; k < x; k++ )  
        m = max( m, v[k] + f( x-k, v ) );  
    return m;  
}
```

¿Cuál es la mejor estructura para el almacén?

Seleccione una:

- ☐ a. int A
- ☒ b. int A[] ✓
- ☐ c. int A[][]

Pregunta 4

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

¿Cuál de estos tres problemas de optimización no tiene, o no se le conoce, una solución voraz óptima?

Seleccione una:

- ☒ a. El problema de la mochila discreta o sin fraccionamiento. ✓
- ☐ b. El árbol de cobertura de coste mínimo de un grafo conexo.
- ☐ c. El problema de la mochila continua o con fraccionamiento.

Pregunta 8

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
unsigned f( unsigned y, unsigned x){ // suponemos y >= x  
    if (x==0 || y==x) return 1;  
    return f(y-1, x-1) + f(y-1, x);  
}
```

¿Cuál es la mejor complejidad espacial que se puede conseguir?

Seleccione una:

- ☐ a. $O(1)$
- ☒ b. $O(y)$ ✓
- ☐ c. $O(y^2)$

Pregunta 11

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

¿Cuál de estas tres estrategias voraces obtiene un mejor valor para la mochila discreta?

Seleccione una:

- ☒ a. Meter primero los elementos de mayor valor específico o valor por unidad de peso. ✓
- ☐ b. Meter primero los elementos de menor peso.
- ☐ c. Meter primero los elementos de mayor valor.

Pregunta 12

Correcta

Puntúa como 1,00

🚩 Marcar pregunta

La solución de programación dinámica iterativa del problema de la mochila discreta ...

Seleccione una:

- ☒ a. ... tiene la restricción de que los valores tienen que ser enteros positivos. ✓
- ☐ b. ... tiene un coste temporal asintótico exponencial con respecto al número de objetos.
- ☐ c. ... calcula menos veces el valor de la mochila que la correspondiente solución de programación dinámica recursiva.

Texto de la pregunta

Cuando se calculan los coeficientes binomiales usando la recursión

$\binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1}$, con $\binom{n}{0} = \binom{n}{n} = 1$, qué problema se da y cómo se puede resolver?

Seleccione una:

- ☐ a. Se repiten muchos cálculos y ello se puede evitar haciendo uso de una estrategia voraz.
- ☒ b. Se repiten muchos cálculos y ello se puede evitar usando programación dinámica.
- ☐ c. La recursión puede ser infinita y por tanto es necesario organizarla según el esquema iterativo de programación dinámica.

Texto de la pregunta

El problema de encontrar el árbol de recubrimiento de coste mínimo para un grafo no dirigido, conexo y ponderado ...

Seleccione una:

- ☐ a. sólo se puede resolver con una estrategia voraz si existe una arista para cualquier par de vértices del grafo.
- ☐ b. ... no se puede resolver en general con una estrategia voraz.
- ☒ c. ... se puede resolver siempre con una estrategia voraz.

Pregunta 8

Correcta

Puntúa como 1,00

⚑ Marcar pregunta

Se pretende implementar mediante programación dinámica recursiva la función recursiva:

```
float f(unsigned x, int y){  
    if( y < 0 ) return 0;  
    float a = 0.0;  
    if ( v1[y] <= x )  
        a = v2[y] + f( x-v1[y], y-1 );  
    float b = f( x, y-1 );  
    return min(a,2+b);  
}
```

¿Cuál es la mejor estructura para el almacén?

Seleccione una:

- ☒ a. unsigned A[][]
- ☐ b. unsigned A
- ☐ c. unsigned A[]

Pregunta 11

Incorrecta

Puntúa como 1,00

⚑ Marcar pregunta

Se pretende implementar mediante programación dinámica iterativa la función recursiva:

```
unsigned f( unsigned x, unsigned v[] ) {  
    if (x==0)  
        return 0;  
    unsigned m = 0;  
    for ( unsigned k = 0; k < x; k++ )  
        m = max( m, v[k] + f( x-k, v ) );  
    return m;  
}
```

¿Cuál es la mejor complejidad espacial que se puede conseguir?

Seleccione una:

- ☒ a. $O(1)$
- ☐ b. $O(x^2)$
- ☐ c. $O(x)$

Dado un problema de optimizacion, el metodo voraz

```
{  
~ Siempre obtiene la solucion optima  
= Garantiza la solucion optima solo para determinados problemas  
~ Siempre obtiene una solucion factible  
}
```

Un tubo de n centimetros de largo se puede cortar en segmentos de 1 centimetro, 2 centimetros, etc. Existe una lista de los precios a los que se venden los segmentos de cada longitud. Una manera de cortar el tubo es la que mas ingresos nos producira. Di cual de estas tres afirmaciones es falsa

```
{  
~ Hacer una evaluacion exhaustiva "de fuerza bruta" de todos las posibles maneras de cortar el tubo consume un tiempo  $\Theta(2^n)$   
= Hacer una evaluacion exhaustiva "de fuerza bruta" de todos las posibles maneras de cortar el tubo consume un tiempo  $\Theta(n!)$   
~ Es posible evitar hacer la evaluacion exhaustiva "de fuerza bruta" guardando, para cada posible longitud  $i < n$  el precio mas elevado posible que se puede obtener diviendo el tubo correspondiente  
}
```