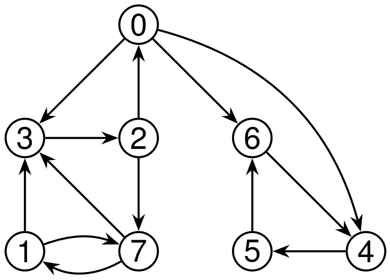
Comenzado el	lunes, 24 de junio de 2024, 15:15
Estado	Finalizado
Finalizado en	lunes, 24 de junio de 2024, 15:59
Tiempo empleado	44 minutos 26 segundos
Puntos	14,17/25,00
Calificación	**5,67** de 10,00 (**56,67**%)

Pregunta 1

Correcta

Se puntúa 1,00 sobre 1,00

¿En qué orden se visitarían los vértices de este grafo dirigido si realizamos un *recorrido en profundidad* del grafo completo? Escribe los identificadores de los vértices separados por espacios en el orden en que son visitados. Supón que los vértices en las listas de adyacentes están ordenados *de mayor a menor*, y que los vértices iniciales que hagan falta también se prueban en ese orden.



Respuesta: 7 3 2 0 6 4 5 1

Los vértices se recorren en este orden: 7 3 2 0 6 4 5 1.

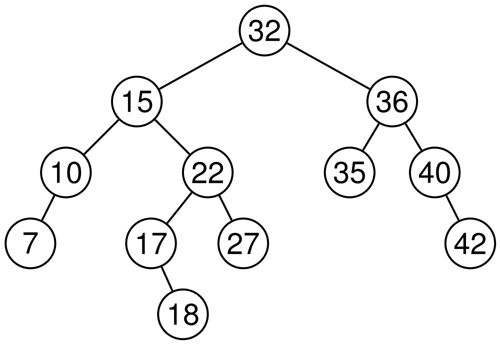
La respuesta correcta es: 7 3 2 0 6 4 5 1

Pregunta 2

Correcta

Se puntúa 1,00 sobre 1,00

Si en este árbol AVL eliminamos el valor 35, ¿qué tipo de rotaciones se producen?



Seleccione una o más de una:

- [x] a. Rotación doble izquierda-derecha. ✔
- [x] b. Rotación simple a la izquierda. ✔
- [ ] c. Rotación simple a la derecha.
- [ ] d. Rotación doble derecha-izquierda.

Tras eliminar el valor 35 el nodo con valor 36 pierde la condición de equilibrio, y hace falta una rotación simple a la izquierda para restablecerla. Eso hace que la raíz también se desequilibre, y haga falta una rotación doble izquierda-derecha para equilibrar el árbol.

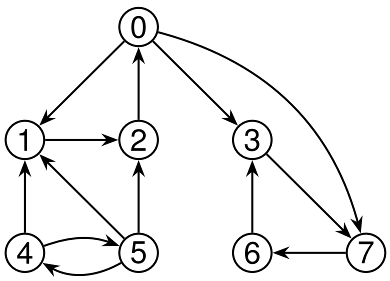
Las respuestas correctas son: Rotación doble izquierda-derecha., Rotación simple a la izquierda.

Pregunta 3

Correcta

Se puntúa 1,00 sobre 1,00

¿En qué orden se visitarían los vértices de este grafo dirigido si realizamos un *recorrido en anchura* del grafo completo? Escribe los identificadores de los vértices separados por espacios en el orden en que son visitados. Supón que los vértices en las listas de adyacentes están ordenados *de mayor a menor*, y que los vértices iniciales que hagan falta también se prueban en ese mismo orden.



Respuesta: 7 6 3 5 4 2 1 0 ✔

Los vértices se recorren en este orden: 7 6 3 5 4 2 1 0.

La respuesta correcta es: 7 6 3 5 4 2 1 0

Pregunta 4

Incorrecta

Se puntúa -0,33 sobre 1,00

En un grafo $G = (V, A)$ donde el coste de un camino es el producto (en lugar de la suma) del coste de sus aristas, queremos calcular las distancias desde un vértice determinado a todos los otros vértices. ¿Cuál de las siguientes condiciones es necesaria y suficiente para que el algoritmo de Dijkstra sea aplicable sin deteriorar su coste?

Seleccione una:

- a. $\text{valor}(a) > 0$ para toda arista $a \in A$ ✖
- b. $\text{valor}(a) > 1$ para toda arista $a \in A$
- c. $\text{valor}(a) \geq 1$ para toda arista $a \in A$
- d. $\text{valor}(a) \geq 0$ para toda arista $a \in A$

El algoritmo de Dijkstra se basa en que añadir una nueva arista $v \to w$ a un camino cv mantiene o aumenta su coste y por tanto es posible expandir una sola vez los vértices del grafo recorriéndolos por distancia creciente al origen con la ayuda de una cola de prioridad. Si c es vacío y v es el vértice inicial, $\text{coste}(vw) = \text{coste}(v) \cdot \text{valor}(v \to w) \geq \text{coste}(v)$ es equivalente a $\text{valor}(v \to w) \geq 1$ porque $\text{coste}(v) = 1$ es un producto vacío. En ese caso, también se cumple $\text{coste}(cvw) = \text{coste}(cv) \cdot \text{valor}(v \to w) \geq \text{coste}(cv)$ para todo camino c.

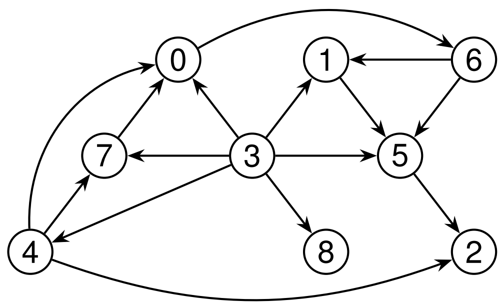
La respuesta correcta es: $\text{valor}(a) \geq 1$ para toda arista $a \in A$

Pregunta 5

Correcta

Se puntúa 1,00 sobre 1,00

Dado el siguiente grafo dirigido,



¿existe un postorden inverso que dé lugar a una ordenación topológica?

Seleccione una:

- a. Verdadero ✔
- b. Falso

Verdadero. El grafo es acíclico y por tanto su postorden inverso (3 8 4 7 0 6 1 5 2) es una ordenación topológica válida.

La respuesta correcta es: Verdadero

Pregunta 6

Incorrecta

Se puntúa -0,33 sobre 1,00

El coste de obtener el mínimo de un AVL con N nodos está en

Seleccione una:

a. $O(N)$

b. $O(1)$ ✖

c. $O(\log N)$

d. $O(N \log N)$

Para buscar el mínimo hay que bajar por el árbol siempre hacia la izquierda hasta que no se pueda más. En el caso peor habría que bajar la altura del árbol, y como el árbol es equilibrado, esta es proporcional al logaritmo del número de nodos.

La respuesta correcta es: $O(\log N)$

Pregunta 7

Finalizado

Sin calificar

Escribe el número del usuario del juez que has utilizado en la primera parte. Por ejemplo, si tu usuario fue TAIS01, escribe el número 1.

Respuesta: 54

La respuesta correcta es: 50

Pregunta 8

Correcta

Se puntúa 1,00 sobre 1,00

La técnica de marcaje en el esquema de ramificación y poda tiene como objetivo:

Seleccione una:

a. determinar de manera más eficiente la factibilidad de una solución parcial. ✔

b. podar un mayor número de nodos.

c. ahorrar espacio.

d. ninguna de las anteriores.

La técnica del marcaje consiste en añadir información adicional en los nodos que permite realizar comprobaciones de factibilidad de forma más eficiente. Determinar la factibilidad de la solución parcial sin los marcadores es igualmente posible pero invirtiendo una mayor cantidad de tiempo, y el número de nodos podados a causa del test de factibilidad será el mismo. El uso de marcadores supone por tanto un mayor gasto de espacio a cambio de una reducción en el tiempo.

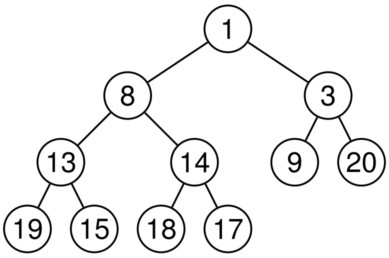
La respuesta correcta es: determinar de manera más eficiente la factibilidad de una solución parcial.

Pregunta 9

Incorrecta

Se puntúa -0,33 sobre 1,00

En este montículo de mínimos, ¿qué valores pueden haber sido el último en insertarse?



Seleccione una:

- a. 8, 14 y 17. ✖
- b. Solamente el 17.
- c. 14 y 17.
- d. Podría ser cualquiera.

El último elemento insertado se añadió en la hoja más a la derecha del último nivel y después fue flotado si hacía falta hacía la raíz. Puede haber sido el 17 y no haber necesitado ser flotado. Puede haber sido el 14 y haberse intercambiado con el 17, que ocupaba su posición. Pero no puede ser el 8, porque entonces el 14 ocuparía su posición antes de la última inserción, y el 14 no puede ser padre del 13 en un montículo de mínimos. Por el mismo motivo no puede ser el 1 el último elemento insertado.

La respuesta correcta es: 14 y 17.

Pregunta 10

Correcta

Se puntúa 1,00 sobre 1,00

Si un grafo dirigido está representado mediante *listas de adyacentes*, ¿cuál es la complejidad del siguiente algoritmo que calcula el número de aristas del grafo si el grafo tiene V vértices y A aristas?

```
Digrafo grafo(V);
int aristas = 0;
for (int v = 0; v < V; ++v)
   aristas += g.ady(v).size();
cout << aristas << '\n';
```

Seleccione una:

- a. $O(V)$ ✔
- b. $O\left(V^2\right)$
- c. $O(V+A)$
- d. $O(V*A)$

Al estar el grafo representado mediante listas de adyacentes, acceder a la lista de adyacentes a un vértice tiene coste constante. En este caso no hace falta recorrer la lista, solamente conocer su longitud, en tiempo constante también. Y eso se hace para cada vértice, por lo que el coste total está en $O(V)$.

La respuesta correcta es: $O(V)$

Pregunta 11

Correcta

Se puntúa 1,00 sobre 1,00

Suponiendo que se dispone de una cantidad inagotable de monedas de euro de todos los valores, ¿cuántas monedas son necesarias para devolver 9.2 euros de forma exacta?

Respuesta: 6

La solución es 6 monedas (4 de 2 euros, 1 de 1 euro y 1 de 20 céntimos).

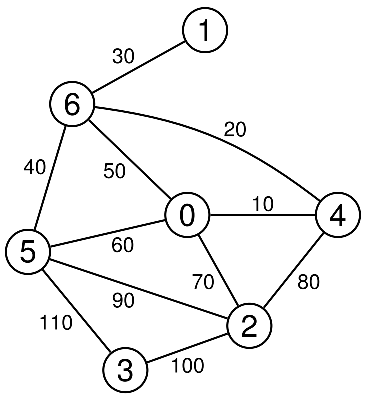
La respuesta correcta es: 6

Pregunta 12

Correcta

Se puntúa 1,00 sobre 1,00

Considera el siguiente grafo no dirigido y valorado,



Da el orden en el que las aristas se añaden al árbol de recubrimiento mínimo (ARM) utilizando el algoritmo de Kruskal (escribe los costes de las aristas, que las identifican, separados por un espacio).

Respuesta: 10 20 30 40 70 100

El algoritmo de Kruskal considera las aristas de menor a mayor coste, y selecciona aquellas que al unirlas a las ya seleccionadas no crean ciclos. Las aristas se añaden al ARM en este orden: 10 20 30 40 70 100.

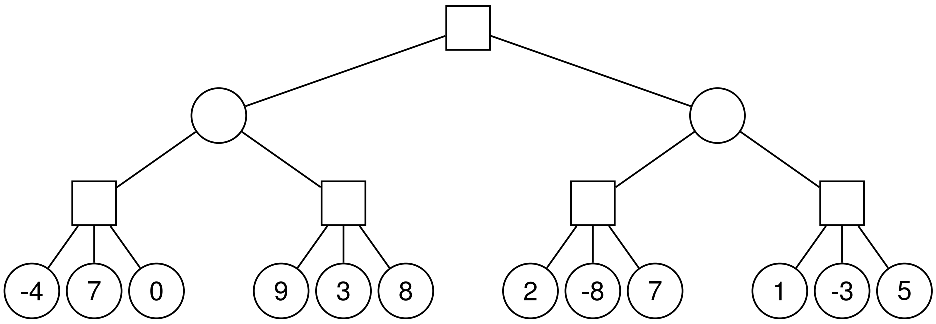
La respuesta correcta es: 10 20 30 40 70 100

Pregunta 13

Sin contestar

Puntúa como 1,00

¿Qué jugada ha de elegir el jugador actual según este árbol de juego y qué valor obtiene con ella?



Seleccione una:

- a. La primera jugada y obtiene valor 9.
- b. Con cualquiera de las dos jugadas obtiene valor 7.
- c. La segunda jugada y obtiene valor 7.
- d. La primera jugada y obtiene valor 7.

El algoritmo minimax calcula desde las hojas hacia la raíz alternando mínimos y máximos. Los valores de los nodos con forma de cuadrado del penúltimo nivel en el dibujo se calculan haciendo el máximo de los valores de sus hijos. De izquierda a derecha esos valores son 7, 9, 7 y 5.

A continuación se calcula el valor de los nodos con forma de círculo haciendo el mínimo de sus hijos, siendo estos valores de izquierda a derecha 7 y 5.

Finalmente, el valor de la raíz es el máximo de estos dos valores. Como 7 > 5, la respuesta correcta es que el jugador debería elegir la primera jugada y obtener valor 7.

Hay otras ramas del árbol de juego que conducen a valores mayores como la que conduce al valor 9, pero si el jugador contrario juega de manera óptima nuestro jugador no tiene opciones de alcanzarla. Si nuestro jugador eligiese la segunda jugada y el jugador contrario jugase de manera óptima como máximo conseguiría el valor 5, aunque si el contrario eligiese mal podría alcanzar igualmente el valor 7.

La respuesta correcta es: La primera jugada y obtiene valor 7.

Pregunta 14

Correcta

Se puntúa 1,00 sobre 1,00

En la facultad de Informática tenemos una sala para tutorías que hay que reservar. Cada persona que quiere usarla pone la hora a la que quiere empezar en la sala y a la hora a la que saldrá. Queremos maximizar el número de usos de la sala en el día y para ello nos planteamos la siguiente estrategia voraz: ordenamos de menor a mayor todas las peticiones por su hora de final y después vamos asignando por orden comprobando que la sala está vacía.

¿Es una estrategia voraz correcta?

Seleccione una:

- a. Verdadero ✔
- b. Falso

Verdadero. Se puede demostrar mediante el método de reducción de diferencias que dicha estrategia es óptima para resolver el problema. Intuitivamente al elegir aquellas peticiones que terminan antes nos aseguramos de que dejamos el suficiente tiempo libre para aceptar el mayor número de peticiones.

La respuesta correcta es: Verdadero

Pregunta 15

Correcta

Se puntúa 1,00 sobre 1,00

Supongamos un sistema monetario con cantidad ilimitada de monedas de valores {2, 5, 10}. Se desea pagar la cantidad 12 con el menor número de monedas. Escribe separados por un espacio y ordenados de menor a mayor los *índices de la última fila* de la matriz rellenada por el algoritmo *que contienen un infinito*.

Respuesta: 1 3 ✔

Las filas se rellenan de arriba abajo y de izquierda a derecha obteniendo la siguiente tabla:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	∞	1	∞	2	∞	3	∞	4	∞	5	∞	6
2	0	∞	1	∞	2	1	3	2	4	3	2	4	3
3	0	∞	1	∞	2	1	3	2	4	3	1	4	2

Por tanto la solución es: 1 3

La respuesta correcta es: 1 3

Pregunta 16

Incorrecta

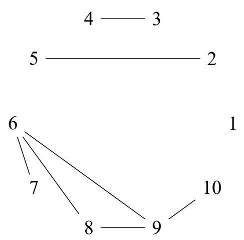
Se puntúa 0,00 sobre 1,00

¿Cuántas componentes conexas tiene el siguiente grafo? El grafo está expresado en el formato utilizado en clase, con los índices de los nodos empezando en 1.

```
10  7
 2  5
 3  4
 6  7
 6  8
 6  9
 8  9
 9 10
```

Respuesta: 3

El resultado es 4.



La respuesta correcta es: 4

Pregunta 17

Sin contestar

Puntúa como 1,00

¿Cuál el resultado de convertir el vector *11 7 13 5 9 16 6 23* en un montículo de mínimos utilizando el algoritmo basado en la operación *hundir*? Escribe el resultado como una lista de números separados por espacios.

Respuesta:

La solución *5 7 6 11 9 16 13 23* se calcula revisando todos los nodos interiores en profundidades decrecientes hasta la raíz, hundiendo aquellos que sean mayores que sus descendientes.

La respuesta correcta es: 5 7 6 11 9 16 13 23

Pregunta 18
Correcta
Se puntúa 1,00 sobre 1,00

Sea G un grafo dirigido con valores positivos en las aristas y $u \to v$ una arista del grafo. Si el camino más corto de s a u tiene un coste de 67 y el camino más corto de s a v vale 57, ¿qué afirmación es cierta?

Seleccione una:

a. $\text{valor}(u \to v) \leq 10$

b. $\text{valor}(u \to v) < 10$

c. $\text{valor}(u \to v) > 10$

d. No se puede decir nada sobre $u \to v$ con esa información. ✔

El camino más corto de s a v no pasa por u porque la distancia de s a u es mayor que la distancia de s a v. El coste de la arista $u \to v$ no afecta por tanto a ninguno de los datos del enunciado y no se puede concluir nada sobre él.

La respuesta correcta es: No se puede decir nada sobre $u \to v$ con esa información.

Pregunta 19
Sin contestar
Puntúa como 1,00

Utilizando las estructuras de partición podemos detectar si un grafo es acíclico. Si están implementadas con *unión rápida por tamaño*, ¿cuál sería la complejidad de dicho algoritmo si el grafo tiene V vértices y A aristas?

Seleccione una:

a. $O(V + A \log V)$

b. $O(A \log A)$

c. $O\left(V + A \lg^* V\right)$

d. Ninguna de las anteriores.

a. Cierto. El algoritmo consiste en recorrer todas las aristas haciendo unión de sus extremos. Si los extremos de una arista ya pertenecen a la misma clase de equivalencia el grafo contiene un ciclo. Se realizan A operaciones de unión sobre V elementos. Si se utiliza union por tamaño (sin compresión de caminos) el coste de cada unión está en $O(\log V)$. Y el coste de todas las operaciones está en $O(V + A \log V)$.
b. Falso. Hay que recorrer todas las aristas haciendo unión de sus extremos. Si los extremos de una arista ya pertenecen a la misma clase de equivalencia el grafo contiene un ciclo. Se realizan A operaciones de unión sobre V elementos. Si se utiliza union por tamaño (sin compresión de caminos) el coste de cada unión está en $O(\log V)$. Y el coste de todas las operaciones está en $O(V + A \log V)$.
c. Falso. Hay que recorrer todas las aristas haciendo unión de sus extremos. Si los extremos de una arista ya pertenecen a la misma clase de equivalencia el grafo contiene un ciclo. Se realizan A operaciones de unión sobre V elementos. Si se utiliza union por tamaño (sin compresión de caminos) el coste de cada unión está en $O(\log V)$. Y el coste de todas las operaciones está en $O(V + A \log V)$.
d. Falso. La respuesta correcta es: $O(V + A \log V)$

La respuesta correcta es: $O(V + A \log V)$

Pregunta 20

Correcta

Se puntúa 1,00 sobre 1,00

El algoritmo de ramificación y poda que resuelve la versión entera del problema de la mochila puede utilizar el algoritmo voraz usado en la versión en que los objetos pueden partirse para asignar la prioridad a los nodos de la cola.

Seleccione una:

- a. Verdadero ✔
- b. Falso

Verdadero. En un problema de maximización la estimación utilizada para asignar prioridad a los nodos ha de ser una cota superior de la mejor solución alcanzable desde ese nodo. Puesto que el valor devuelto por la estrategia voraz es una cota superior de cualquier solución que respete la restricción de peso límite del problema, al aplicarla a partir de una solución parcial obtenemos una cota superior de todas las soluciones alcanzables a partir de ella.

La respuesta correcta es: Verdadero

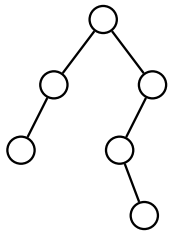
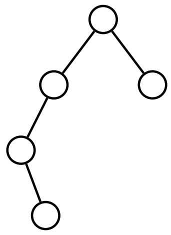
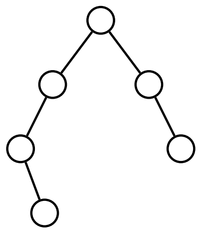
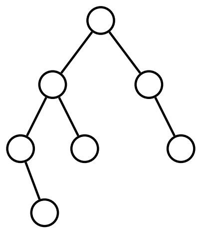
Pregunta 21

Correcta

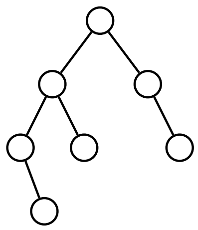
Se puntúa 1,00 sobre 1,00

¿Cuáles de los siguientes árboles binarios están equilibrados?

Seleccione una o más de una:

- [ ] a. 
- [ ] b. 
- [ ] c. 
- [x] d.  ✔ Cierto. Todos los nodos cumplen la condición de equilibrio.

a. Falso. El hijo derecho de la raíz no cumple la condición de equilibrio, al tener un hijo izquierdo de altura 2 y un hijo derecho de altura 0.
b. Falso. La raíz no cumple la condición de equilibrio, al tener un hijo izquierdo de altura 3 y un hijo derecho de altura 1.
c. Falso. El hijo izquierdo de la raíz no cumple la condición de equilibrio, al tener un hijo izquierdo de altura 2 y un hijo derecho de altura 0.
d. Cierto. Todos los nodos cumplen la condición de equilibrio.

La respuesta correcta es: 

Pregunta 22

Correcta

Se puntúa 1,00 sobre 1,00

Un grafo dirigido admite una ordenación topológica si y solo si todas sus componentes fuertemente conexas son conjuntos unitarios.

Seleccione una:

- a. Verdadero ✔
- b. Falso

Verdadero. Un grafo dirigido es acíclico si y solo si todas sus componentes fuertemente conexas son unitarias (si hubiera un ciclo no trivial, todos los vértices del ciclo estarían en la misma componente conexa, que no sería unitaria). Por otro lado, un grafo dirigido admite una ordenación topológica si y solo si es acíclico.

La respuesta correcta es: Verdadero

Pregunta 23

Incorrecta

Se puntúa -0,50 sobre 1,00

Para demostrar que una estrategia voraz no es válida basta encontrar un contraejemplo donde no devuelva la solución óptima.

Seleccione una:

- a. Verdadero
- b. Falso ✖

Cierto. Si la estrategia no encuentra la solución óptima en un caso particular es claro que no resuelve el problema en general.

La respuesta correcta es: Verdadero

Pregunta 24

Correcta

Se puntúa 1,00 sobre 1,00

El coste del algoritmo que ordena N valores insertándolos primero en un AVL y recorriéndolo completamente a continuación para producir la lista ordenada de dichos elementos está en el caso peor en

Seleccione una:

- a. $O(\log N)$
- b. $O(N \log N)$ ✔
- c. $O(N^2)$
- d. $O(N)$

Al mantenerse el árbol equilibrado, cada inserción tiene un coste que está acotado superiormente por el logaritmo de N. Por tanto la construcción del árbol tiene un coste en $O(N \log N)$. Producir la lista ordenada requiere un recorrido en inorden, con un coste en $O(N)$.

La respuesta correcta es: $O(N \log N)$

Pregunta 25

Correcta

Se puntúa 1,00 sobre 1,00

El problema del cambio de monedas de una cantidad C con un número ilimitado de monedas de n tipos, resuelto por programación dinámica, puede tener un coste en espacio adicional en $\Theta(C)$ incluso cuando queremos saber cuántas monedas de cada tipo utilizar.

Seleccione una:

- a. Verdadero ✔
- b. Falso

Verdadero. En el problema del cambio de monedas, al contrario que en otros problemas, toda la información necesaria para reconstruir la solución se encuentra en la última fila debido a que el número de monedas de cada tipo es ilimitado. Por ello, se puede reconstruir la solución incluso cuando optimizamos el espacio usando del orden de $\Theta(C)$ memoria adicional.

La respuesta correcta es: Verdadero

Pregunta 26

Incorrecta

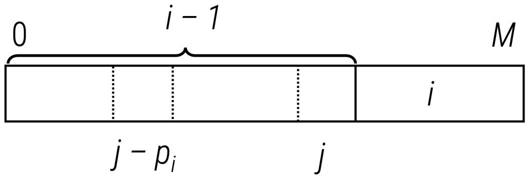
Se puntúa -0,33 sobre 1,00

Si queremos reducir el coste en espacio del algoritmo de programación dinámica que resuelve la versión entera del problema de la mochila (cuando solamente estamos interesados en el valor óptimo y no en los objetos que hay que meter en la mochila) utilizando un vector en lugar de una matriz, debemos rellenar el vector:

Seleccione una:

- a. no se puede reducir el coste en espacio.
- b. de izquierda a derecha.
- c. tanto de derecha a izquierda como de izquierda a derecha. ✖
- d. de derecha a izquierda.

En cada iteración del algoritmo en que consideramos el objeto i-ésimo rellenamos un vector de $M + 1$ posiciones, siendo M el peso límite de la mochila. Al empezar la iteración i-ésima, el vector contiene los valores correspondientes al objeto $i - 1$. Puesto que para calcular los valores correspondientes al objeto i y cada peso límite j necesitamos dos valores correspondientes al objeto $i - 1$ en columnas j y $j - p_i$ (siendo p_i el peso del objeto i), se ha de rellenar el vector de *derecha a izquierda* para mantener dichos valores disponibles hasta realizar el cálculo, tal como se muestra en el siguiente dibujo:



La respuesta correcta es: de derecha a izquierda.