PROBLEMAS EXAMEN

DIJKSTRA, PRIM, KRUSKAL

F1 2018 A

4. Sobre al algoritmo de Dijkstra, cuál de las siguientes afirmaciones es **falsa**:

   (a) El algoritmo de Dijkstra sigue el esquema voraz.
   (b) El algoritmo de Dijkstra determina la longitud del camino de coste, peso o distancia mínima que va desde el nodo origen a cada uno de los demás nodos del grafo.
   (c) El coste del algoritmo de Dijkstra es O(n log n).
   (d) En el algoritmo de Dijkstra un camino desde el nodo origen hasta otro nodo es *especial* si se conoce el camino de coste mínimo desde el nodo origen a todos los nodos intermedios.

F2 2018 A

5. Sea un grafo denso no dirigido representado con la siguiente matriz de adyacencia, en la que puede haber pesos de valor 0:

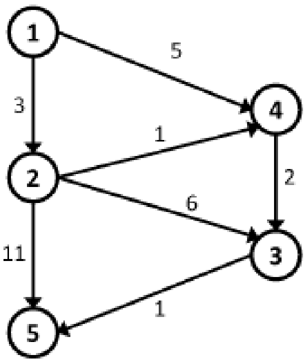
	A	B	C	D	E	F
A	-	8	3	0	0	0
B		-	5	0	0	2
C			-	1	4	0
D				-	7	0
E					-	6
F						-

Si se utiliza el algoritmo de Prim para calcular un árbol de recubrimiento mínimo tomando como raíz del árbol el nodo A, indique cuál de las siguientes secuencias de aristas representa el orden en el que las selecciona el algoritmo de Prim como integrantes del árbol de recubrimiento mínimo:

(a) {A,C},{C,D},{C,E},{B,C},{B,F}.
(b) {A,C},{C,D},{B,F},{C,E},{B,C}.
(c) {A,C},{C,D},{C,E},{B,F},{B,C}.
(d) Ninguna de las anteriores.

SO 2018

4. Dado el siguiente grafo, indique cuáles serían, de acuerdo al algoritmo de Dijkstra, las longitudes de los caminos de coste mínimo que unen el nodo 1 con el resto de nodos del grafo:



(a) {3,4,6,7}
(b) {3,5,7,8}
(c) {3,6,5,7}
(d) Ninguna de las anteriores

SR 2018

4. Aplicando el algoritmo de Dijkstra al grafo con conjunto de vértices V={1, 2, …, 8} y aristas con pesos (entre paréntesis) representadas por la lista de adyacencia:

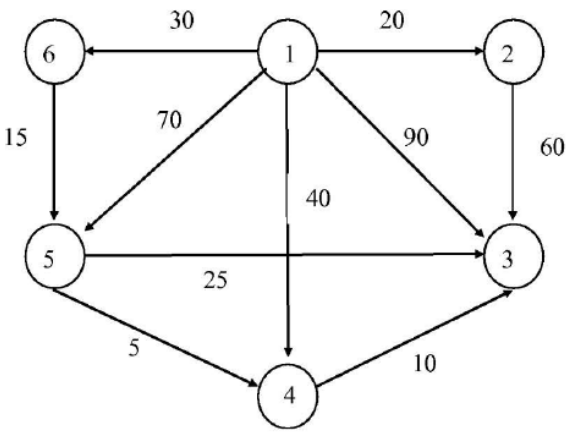
1 → 2(4), 3(10), 4(2), 6(8)
2 → 4(4)
3 → 6(1), 8(10)
4 → 5(2), 3(1), 7(5)
5 → 3(5), 6(1)
6 → 3(9)
7 → 1(7)
8 → 3(2), 4(3), 7(2)

Indicar cuál de las siguientes **NO** es un valor correcto para el vector "Especial" D[i] generado por el algoritmo de Dijkstra a lo largo de las iteraciones, representando en cada iteración la distancia mínima hasta el momento entre los vértices 1 e i:

(a) D = [4,3,2,4,8,7, ∞]
(b) D = [4,10,2, ∞,8, ∞,∞]
(c) D = [4,3,2,4,4,7,13]
(d) D = [4,9,2, 4,4, 7,∞]

F1 2017 A

4. Dado el grafo de la figura:

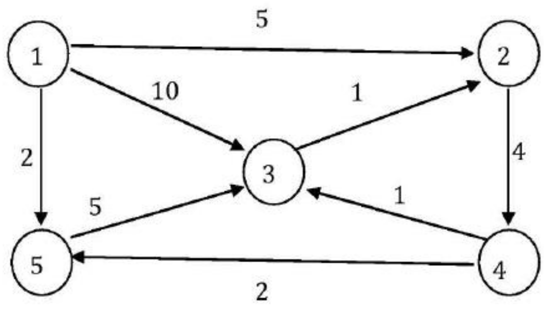


Indique cuál sería el orden en que se seleccionan los nodos del conjunto de candidatos al aplicar el algoritmo de Dijkstra comenzando por el nodo 1:

(a) 1 2 6 5 3 4
(b) 1 2 6 5 4 3
(c) 1 2 6 4 5 3
(d) Ninguna de las anteriores

F2 2017 A

5. Aplicando el algoritmo de Dijkstra al grafo:



Indicar cuál de las siguientes **NO** es un valor correcto para el vector "Especial" D[i] generado por el algoritmo de Dijkstra a lo largo de las iteraciones, representando en cada iteración la distancia mínima hasta el momento entre los vértices 1 e i:

(a) D = [5,10,∞,2]
(b) D = [5,7,∞,2]
(c) D = [5,10,9,2]
(d) D = [5,7,9,2]

SO 2017

1. Sea un grafo denso no dirigido representado con la siguiente matriz de adyacencia, en la que puede haber pesos de valor 0:

	1	2	3	4	5	6	7	8
1	-	2	3	4	5	0	1	2
2		-	0	2	4	0	2	4
3			-	0	3	0	3	0
4				-	2	0	4	2
5					-	0	5	4
6						-	0	0
7							-	2
8								-

Si se utiliza el algoritmo de Prim para calcular un árbol de recubrimiento mínimo tomando como raíz del árbol el nodo 1, indica cuál de las siguientes secuencias de aristas representa el orden en el que las selecciona el algoritmo de Prim como integrantes del árbol de recubrimiento mínimo:

a. {1,6},{1,7},{6,2},{6,3},{4,6},{6,5},{6,8}.
b. {1,7},{1,6},{6,3},{6,2},{4,6},{6,5},{6,8}.
c. {1,6},{6,2},{2,3},{3,8},{4,6},{6,5},{6,7}.
d. {1,6},{1,2},{2,3},{4,6},{6,5},{6,8},{6,7}.

SR 2017

2. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado con programación dinámica. Supongamos que se dispone de 5 objetos con pesos: 1,3,4,5,7 y beneficios: 2,5,10,14,15 respectivamente, y un volumen máximo de 8. Identifica cuál de las siguientes respuestas correspondería al contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 5, si dichos objetos se consideran en el orden indicado.

    a. 0 2 2 5 7 7 7 7 7
    b. 0 2 2 5 10 12 12 15 15
    c. 0 2 2 5 10 12 14 16 19
    d. 0 2 2 5 10 14 16 16 19

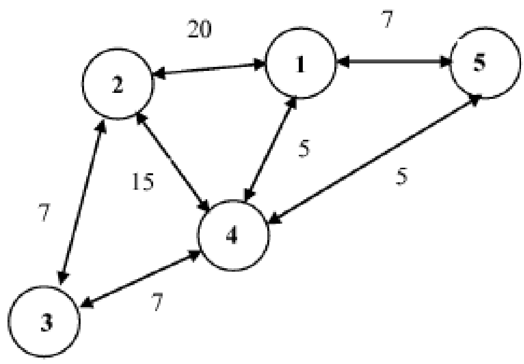
3. Sea el algoritmo de Dijkstra. Indica cuál de las afirmaciones siguientes <u>es cierta</u> en relación a su coste.

    a. El coste del algoritmo es independiente de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen.

b. El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. Pero en el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo es independiente de que el grafo sea disperso o denso.
c. El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. Pero en el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo si el grafo es disperso está en $O(n^2 \log n)$.
d. Ninguna de las afirmaciones anteriores es cierta.

SR 2016 A

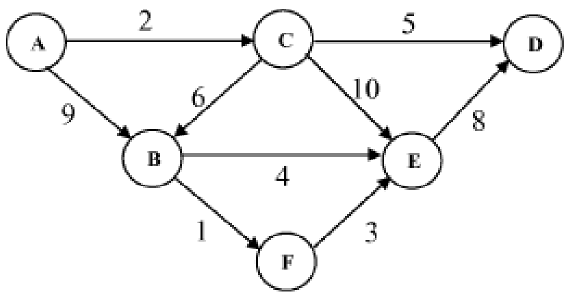
5. Dado el siguiente grafo al que se le aplica el algoritmo de Dijkstra tomando como nodo origen el 1, detallar la evolución del vector de distancias tras haber seleccionado el algoritmo el nodo 3. *NOTA: el vector especial (de distancias) definido entre [2..n] contiene la distancia mejor hasta el momento entre el nodo 1 y los demás.*



(a) [20,∞,5,7]
(b) [20,12,5,7]
(c) [19,∞,5,7]
(d) [19,12,5,7]

F1 2015 A

2. Dado el grafo de la siguiente figura:



indicad cuál sería el orden en que se seleccionarían (pasan a estar explorados) los nodos al aplicar el algoritmo de Dijkstra desde el nodo A:

(a) A,C,D,B,F,E
(b) A,F,C,E,B,D
(c) A,C,B,F,E,D
(d) Ninguna de las anteriores

F2 2015 A

3. Aplicando el algoritmo de Dijkstra al grafo con conjunto de vértices {1,2,…8} y aristas con pesos (entre paréntesis) representadas por lista de adyacencia:

   1 → 2(4), 3(10), 4(2), 6(8)
   2 → 4(4)
   3 → 6(1), 8(10)
   4 → 5(2), 6(1), 7(5)
   5 → 3(5), 6(1)
   6 → 3(9)
   7 → 1(7)
   8 → 3(2), 4(3), 7(2)

Cuál de las siguientes NO es un valor correcto para el vector "Especial" $D[i]$ generado por el algoritmo de Dijkstra a lo largo de las iteraciones, representando en cada iteración la distancia mínima hasta el momento entre los vértices 1 e i:

(a) D = [4,10,2, ∞,8, ∞,∞]
(b) D = [4,10,2, 4,3, 7,∞]
(c) D = [4,9,2, 4,3, 7,∞]
(d) D = [4,10,2, 4,3, 7,19]

6. Sea un grafo denso no dirigido representado con la siguiente matriz de adyacencia, en la que puede haber pesos de valor 0:

	1	2	3	4	5	6	7	8
1	-	2	3	4	5	0	1	2
2		-	0	2	4	0	2	4
3			-	0	3	0	3	0
4				-	2	0	4	2
5					-	0	5	4
6						-	0	0
7							-	2
8								-

Si se utiliza el algoritmo de Prim para calcular un árbol de recubrimiento mínimo tomando como raíz del árbol el nodo 1, indica cuál de las siguientes secuencias de aristas representa el orden en el que las selecciona el algoritmo de Prim como integrantes del árbol de recubrimiento mínimo:

SO 2015 A

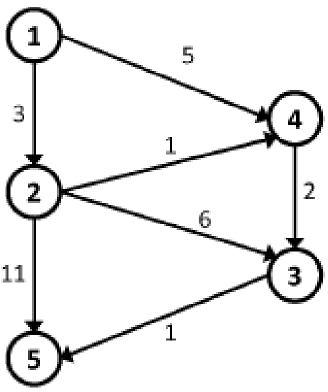
2. Sea un grafo no dirigido representado con la siguiente matriz de adyacencia:

	1	2	3	4	5	6	7
1	-	10			6		
2		-	5	2			
3			-	4			
4				-		3	
5					-	9	1
6						-	8
7							-

Si se utiliza el algoritmo de Kruskal para calcular un árbol de recubrimiento mínimo, indica cuál de las siguientes secuencias de aristas (que incluye las rechazadas) representa el orden en el que las evalúa el algoritmo:

(a) {5,7},{2,4},{4,6},{3,4},{2,3},{1,5},{6,7}.
(b) {5,7},{2,4},{4,6},{3,4},{1,5},{6,7},{5,6}.
(c) {5,7},{1,5},{6,7},{4,6},{2,4},{4,3}.
(d) {1,5},{5,7},{6,7},{4,6},{2,4},{4,3}.

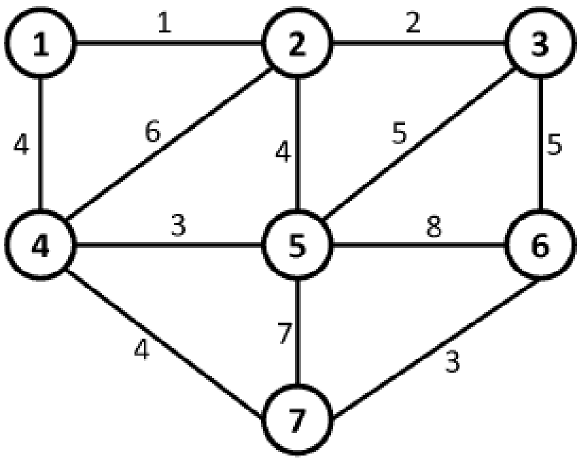
5. Dado el siguiente grafo, indique cuál sería el orden en que se seleccionarían los nodos (pasan a estar explorados) al aplicar el algoritmo de Dijkstra para encontrar todos los caminos de menor coste desde el nodo 1:



(a) {1, 2, 4, 3, 5}
(b) {1, 4, 3, 5, 2}
(c) {1, 2, 3, 4, 5}
(d) Ninguna de las anteriores

SR 2015 A

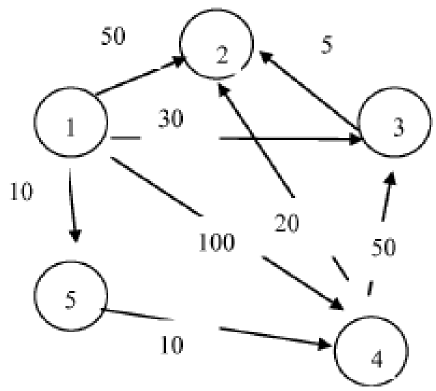
6. Dado el siguiente grafo no dirigido, aplique el algoritmo de Kruskal para hallar el árbol de recubrimiento mínimo e indique cuál de las siguientes afirmaciones es **cierta**:



(a) El coste del algoritmo empleado es Θ (n^2).
(b) Un orden válido en el que las aristas se incluyen en el árbol de recubrimiento es {{1,2} {2,3} {4,5} {6,7} {1,4} {4,7}}.
(c) Un orden válido en el que las aristas se incluyen en el árbol de recubrimiento es {{1,2} {2,3} {6,7} {4,5} {1,4} {2,5}}.
(d) Ninguna de las anteriores es correcta.

F1 2014 A

2. Dado el siguiente grafo dirigido:

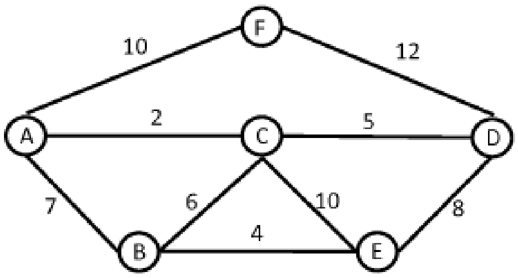


Indique cuál sería el valor del vector *especial[]* en el primer y penúltimo paso del algoritmo de Dijkstra:

(a) [50,30,100,10] y [35,30,20,10]
(b) [50,30,100,10] y [40,30,20,10]
(c) [50,30,100,10] y [35,30,100,10]
(d) Ninguna de las anteriores

F2 2014 A

5. Dado el siguiente grafo no dirigido:

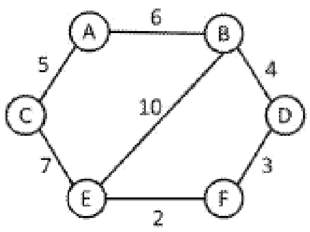


Indique cuál sería el orden en que se seleccionarían las aristas al aplicar el algoritmo de Kruskal:

(a) {{A,C},{C,D},{C,B},{B,E},{D,F}}
(b) {{A,C},{B,E},{C,D},{C,B},{A,F}}
(c) {{A,C},{C,D},{A,B},{B,E},{A,F}}
(d) Ninguna de las anteriores

SO 2014 A

4. Sea el grafo de la figura:

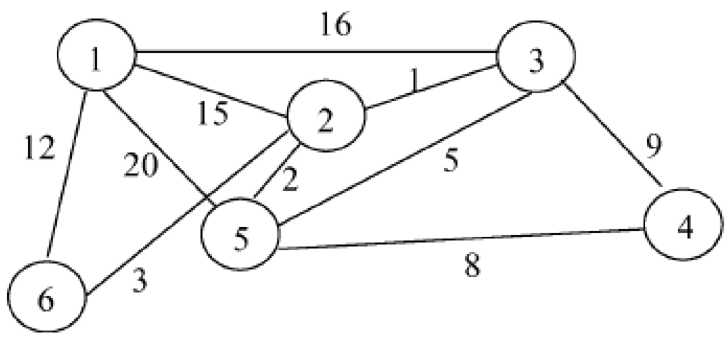


Indique cuál sería el orden en que se seleccionan los nodos del conjunto de candidatos al aplicar el algoritmo de Dijkstra comenzando por el nodo A:

(a) A C B D E F
(b) A B C D E F
(c) A B D C F E
(d) Ninguna de las anteriores

SR 2014 A

1. Dado el grafo no dirigido de la figura:



Indica cuál sería el orden en que se seleccionarían (pasan a pertenecer al árbol) las aristas al aplicar el algoritmo de Kruskal:

(a) (2,3)(3,5)(5,4)(2,6)(6,1)
(b) (1,6)(6,2)(2,3)(3,4)(4,5)
(c) (2,3)(2,5)(2,6)(5,4)(1,6)
(d) Ninguna de las anteriores es cierta

3. Sea el algoritmo de Dijkstra. Indica cuál de las afirmaciones siguientes es **cierta** en relación a su coste.

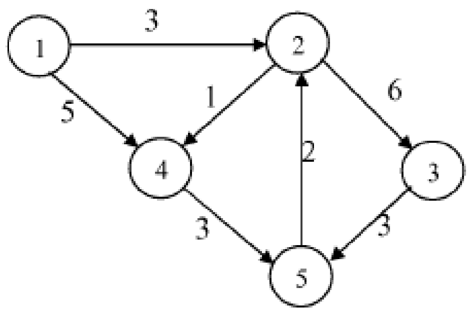
(a) El coste del algoritmo es independiente de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen.
(b) El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. En el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo es independiente de que el grafo sea disperso o denso.
(c) El coste del algoritmo sí depende de cómo se implemente el array *especial*[] que almacena las distancias mínimas desde cada nodo al origen. En el caso de que la implementación de *especial*[] sea mediante un montículo de mínimos y el grafo sea conexo, el coste del algoritmo varía dependiendo de que el grafo sea disperso o denso.
(d) Ninguna de las afirmaciones anteriores es cierta.

F2 2013 A

2. Los algoritmos de Prim y Kruskal son algoritmos voraces que se aplican a la hora de calcular un árbol de recubrimiento mínimo. En relación a ambos algoritmos, cuál de las siguientes afirmaciones es cierta.

a) Al inicializar el algoritmo Prim selecciona un nodo arbitrario mientras Kruskal elige aquel nodo con menor número de aristas.
b) Prim parte de un nodo arbitrario, mientras Kruskal parte del conjunto de aristas ordenadas de menor a mayor coste.
c) Prim es más eficiente en el caso de un grafo muy disperso, mientras Kruskal es más eficiente si el grafo es muy denso.
d) Ninguna de las anteriores.

6. Dado el grafo de la siguiente figura:

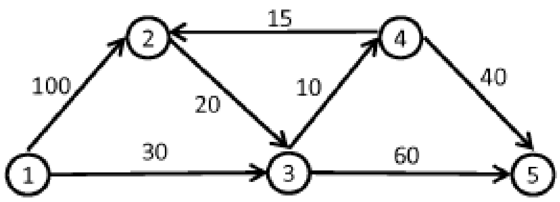


indicar cuál sería el orden en que se seleccionarían (pasan a estar explorados) los nodos al aplicar el algoritmo de Dijkstra desde el nodo 1:

a) 1,2,4,5,3.
b) 1,2,5,4,3.
c) 1,2,3,5,4.
d) Ninguna de las anteriores.

SO 2013 A

4. Dado el siguiente grafo dirigido:

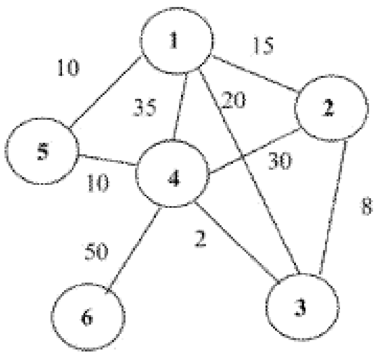


Indique el valor del vector de distancias *especial[]* en el paso del algoritmo de Dijkstra en el que se selecciona el nodo *v=4*, tomando como nodo origen el nodo 1:

a) [∞, 30,40, ∞]
b) [50, 30,40, ∞]
c) [55, 30,40, 80]
d) Ninguna de las anteriores

SR 2013 A

5. Dado el siguiente grafo:

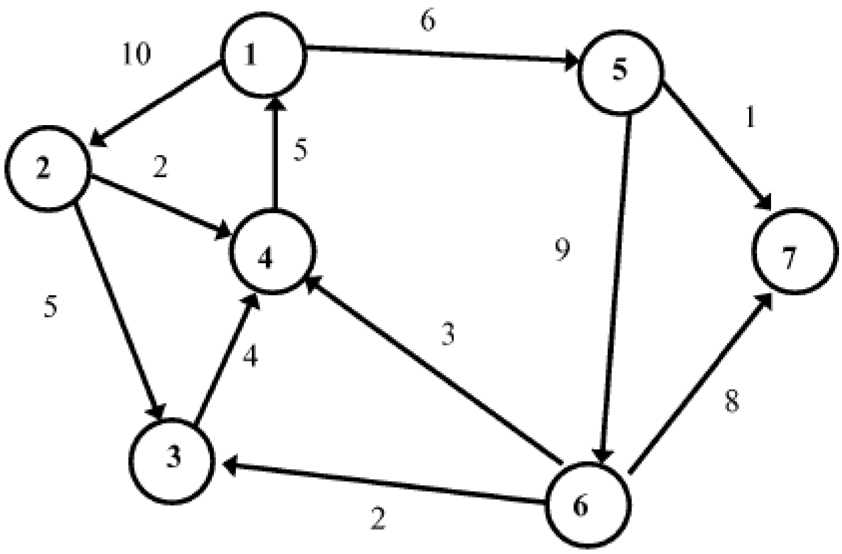


Indique cuál de las siguientes opciones se corresponde con el orden en el que se incluyen los nodos en el árbol de recubrimiento mínimo aplicando el algoritmo de Prim y empezando por el nodo 1.

a) 1,5,2,6,3,4
b) 1,2,3,4,5,6
c) 1,5,4,3,2,6
d) Ninguno de los anteriores

F1 2012 A

3.- Dado el siguiente grafo dirigido:

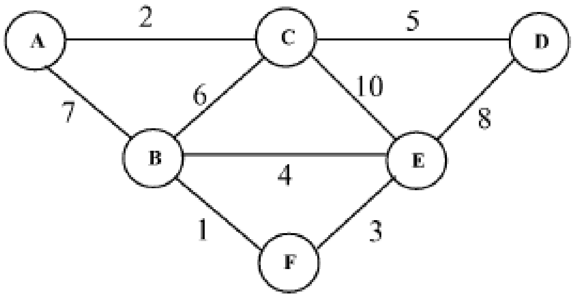


Se pide detallar el valor del vector de distancias *especial[]* en el paso del algoritmo de Dijkstra en el que se selecciona el nodo v=7, tomando como nodo origen el nodo 1.

a. [10,15,12,6,15,7]
b. [10, ∞, ∞,6,15,7]
c. [10,∞,∞,6,∞,∞]
d. Ninguna de las anteriores.

F2 2012 A

4. Dado el grafo no dirigido de la figura,



indique cuál sería el orden en que se seleccionarían (pasan a pertenecer al árbol) los nodos al aplicar el algoritmo de Prim comenzando por el nodo A:

b. A, F, E, C, B, D
c. A, B, C, E, D, F
d. A, C, D, B, F, E
e. Ninguno de los anteriores

SR 2012 A

6. En el cálculo del coste del algoritmo de Prim:
   a. El hecho de que el grafo sea denso o disperso no afecta al coste cualquiera que sea su implementación.
   b. La implementación mediante matriz de adyacencia es ineficiente si el grafo es disperso.
   c. La implementación mediante listas de adyacencia y montículo para representar a los candidatos pendientes es siempre cuadrática.
   d. Ninguna de las anteriores es cierta.

QUICKSORT

F2 2018 A

3. Indica cuál de las siguientes afirmaciones es cierta con respecto al algoritmo Quicksort para un vector de tamaño n.

(a) Se compone de dos invocaciones recursivas con tamaño n/2 más un procedimiento que combina los subvectores ordenados resultantes que es de coste lineal.
(b) Su coste en el caso peor es $O(n^2)$, pero existe una versión mejorada eligiendo como pivote la mediana del vector, lo que daría que el coste del Quicksort en el caso peor sería O(n).
(c) Está compuesto de dos recorridos lineales del vector y posteriormente una llamada recursiva al subvector no ordenado.
(d) Ninguna de las anteriores es cierta.

SR 2018

5. Dado un algoritmo de Quicksort que elige como pivote el primer elemento del vector, contestar cuál es la opción **cierta**:

(a) El vector [1,2,3,4,5] es un ejemplo de caso peor.
(b) El vector [3,1,5,2,4] es un ejemplo de caso peor.
(c) El vector [5,4,3,2,1] es un ejemplo de caso mejor.
(d) El vector [1,2,3,4,5] se ordenaría con coste O (n log n).

SO 2017

4. Considérese el vector [7,10,5,2,20,15,1,5]. Los vectores argumento de la primera invocación recursiva del algoritmo de quicksort, cuando se toma el elemento 7 de la primera posición como pivote, son:

a. [5,2,5,1] y [15,20,10]
b. [1,5,5,2] y [15,20,10]
c. [5,2,5,1] y [10,20,15]
d. Ninguna de las opciones anteriores.

SO 2016

3. El algoritmo Quicksort tiene:

(a) Caso medio de orden $O(n^2)$
(b) Caso peor de orden $O(n^2 \log n)$
(c) Caso mejor de orden O(n)
(d) Caso mejor de orden O(n log n)

SR 2016

3. Dado un algoritmo de Quicksort que elige como pivote el primer elemento del vector, contestar cuál es la opción **falsa**:

   (a) El vector [1,2,3,4,5] es un ejemplo de caso peor.
   (b) El vector [3,1,5,2,4] es un ejemplo de caso medio.
   (c) El vector [5,4,3,2,1] es un ejemplo de caso mejor.
   (d) El vector [1,2,3,4,5] se ordenaría con coste cuadrático.

F1 2015 A

1. Considérese el vector [5, 2, 7, 3, 1, 8, 2, 6, 9]. Los vectores argumento de la primera invocación recursiva del algoritmo de quicksort, cuando se toma el elemento 5 de la primera posición como pivote, son:

   (a) [2,3,1,2] y [8,7,6,9]
   (b) [2,7,3,1] y [8,2,6,9]
   (c) [1,2,2,3] y [8,7,6,9]
   (d) Ninguna de las opciones anteriores.

F2 2015 A

4. Dado el vector M= [5, 6, 12, 1, 7, 9, 3, 7, 1, 11, 3, 2, 8, 6, 5, 9] al que aplicamos el algoritmo de ordenación Quicksort tomando como pivotes los valores M[0] y para una ordenación de menor a mayor. Se pide contestar la opción verdadera:

   (a) La primera recursión del algoritmo genera dos llamadas a quicksort con argumentos [3 5 2 1 3 1] y [7 9 11 7 12 8 6 6 9] con pivote M[0] = 5.
   (b) La primera recursión del algoritmo genera dos llamadas a quicksort con argumentos [6 12 1 7 9 3 7] y [1 11 3 2 8 6 5 9] con pivote M[0] = 5.
   (c) La primera recursión del algoritmo genera dos llamadas a quicksort con argumentos [1 1 2 3 3 5] y [6 6 7 7 8 9 9 11 12] con pivote M[0] = 5.
   (d) La primera recursión del algoritmo genera dos llamadas a quicksort con argumentos [1 3 6 7 7 9 12] y [1 2 3 5 6 8 9 11] con pivote M[0] = 5.

SR 2105 A

4. ¿Cuál de las siguientes afirmaciones es **falsa**?

   (a) El coste del algoritmo de ordenación mergesort es $\Theta(n \log n)$.
   (b) El algoritmo de ordenación Quicksort en el caso peor tiene coste $\Theta(n \log n)$.
   (c) El algoritmo de ordenación basado en montículos tiene coste $\Theta(n \log n)$.
   (d) Para que el algoritmo Quicksort sea eficiente, la descomposición en subproblemas tiene que ser equilibrada.

SR 2013

6. Indique cuál de las siguientes afirmaciones es cierta con respecto al algoritmo Quicksort de un vector de tamaño n.

   a) Está compuesto de dos invocaciones recursivas con tamaño n/2 más una parte posterior que combina los subvectores ordenados resultantes que es de coste lineal.
   b) Está compuesto de dos recorridos lineales al vector y posteriormente una llamada recursiva al subvector no ordenado.
   c) Se suele utilizar una operación de orden O(1) para hallar el pivote y luego dos llamadas recursivas que en general no son de tamaño n/2.
   d) Ninguna de las anteriores es cierta.

F2 2012 A

2. Dado el vector v=[5,6,9,3,4,1] y el algoritmo Quicksort, los vectores argumento de la primera invocación recursiva, con pivote v[1] = 5, son:

   a. [3,4,1] y [6,9]
   b. [5,6,9] y [5,3,4,1]
   c. [5] y [6,9,3,4,1]
   d. [1] y [5,6,9,3,4]

SR 2012

3. Se desea implementar el algoritmo de ordenación rápida (quicksort) para aplicarlo a vectores que están casi ordenados. A la hora de elegir el elemento pivote para la partición de los subvectores, ¿Cuál sería la elección más adecuada para este caso concreto?
   a. El primer elemento del subvector.
   b. El último elemento del subvector.
   c. El elemento que se encuentra en la posición media del subvector.
   d. La elección del elemento pivote no influye en el rendimiento del algoritmo.

ALGORITMO DEVOLUCION MONEDAS

F2 2018 A

2. Indica cuál de las siguientes afirmaciones es cierta con respecto al problema de la devolución de cambio de moneda utilizando un número mínimo de monedas y suponiendo que la disponibilidad de cada tipo de moneda es ilimitada:

   (a) El problema de la devolución de cambio de moneda se puede resolver para todo sistema monetario utilizando una estrategia voraz, siempre que en el sistema se disponga de un tipo de moneda de 1.
   (b) Si se dispone de n tipos de moneda $T = \{m^0, m^1, m^2, \ldots m^{n-1}\}$ siendo $m > 1$ y $n > 0$, el problema de la devolución de cambio de moneda no se puede resolver utilizando una estrategia voraz.
   (c) El problema de la devolución de cambio de moneda para cualquier sistema monetario, siempre que en el sistema se disponga de un tipo de moneda de 1, se puede resolver utilizando los esquemas de programación dinámica y de ramificación y poda.
   (d) El problema de la devolución de cambio de moneda para cualquier sistema monetario, siempre que en el sistema se disponga de un tipo de moneda de 1, se puede resolver utilizando el esquema de programación dinámica pero no el de ramificación y poda.

F1 2017 A

2. Sea el problema de la devolución de cambio con monedas de valores 1,6 y 10 solucionado con programación dinámica para pagar una cantidad de 12 unidades. Identifica cuál de las siguientes respuestas correspondería al contenido de la tabla de resultados parciales de cantidades en la fila correspondiente a la moneda de valor 6, si dichas monedas se consideran por orden creciente de valores:

   (a) 0 1 2 3 4 5 6 2 3 4 5 6 3
   (b) 0 1 2 3 4 5 6 2 3 4 5 6 7
   (c) 0 1 2 3 4 5 1 2 3 4 5 6 2
   (d) Ninguna de las anteriores.

F1 2015 A

5. Dado el problema de la devolución del cambio para una cantidad C>0. Indica cuál de estas afirmaciones es **cierta**.

   (a) El esquema de ramificación y poda es el único que puede resolver de forma óptima el problema cuando se dispone de un conjunto finito de tipos de moneda $T = \{m^0, m^1, m^2, \ldots m^n\}$, siendo $m > 1$ y $n>0$.
   (b) Se puede encontrar una estrategia voraz que permita resolver el problema de forma óptima para cualquier conjunto de monedas.
   (c) El esquema de programación dinámica puede resolver de forma óptima el problema cuando no se cumpla que $T = \{m^0, m^1, m^2, \ldots m^n\}$, siendo $m > 1$ y $n>0$.
   (d) El esquema de vuelta atrás es el más apropiado para resolver este problema cuando se dispone de un conjunto finito de tipos de moneda $T = \{m^0, m^1, m^2, \ldots m^n\}$, siendo $m > 1$ y $n>0$.

F2 2013 A

4. Indica cuál de las siguientes afirmaciones es cierta con respecto al problema de la devolución de cambio de moneda utilizando un número mínimo de monedas y suponiendo que la disponibilidad de cada tipo de moneda es ilimitada:

   a) El problema de la devolución de cambio de moneda se puede resolver para todo sistema monetario utilizando una estrategia voraz, siempre que en el sistema se disponga de un tipo de moneda de 1.
   b) Si se dispone de n tipos de moneda $T = \{m^0, m^1, m^2, \dots m^n\}$ siendo $m > 1$ y $n > 0$, el problema de la devolución de cambio de moneda no se puede resolver utilizando una estrategia voraz.
   c) El problema de la devolución de cambio de moneda para cualquier sistema monetario, siempre que en el sistema se disponga de un tipo de moneda de 1, se puede resolver utilizando los esquemas de programación dinámica y de ramificación y poda.
   d) El problema de la devolución de cambio de moneda para cualquier sistema monetario, siempre que en el sistema se disponga de un tipo de moneda de 1, se puede resolver utilizando el esquema de programación dinámica pero no el de ramificación y poda.

SO 2013

6. Se dispone de cuatro tipos de monedas de valores 1, 2, 4 y 8. Se desea resolver el problema de pagar una cantidad C > 0 utilizando un número mínimo de monedas y suponiendo que la disponibilidad de cada tipo de moneda es ilimitada. ¿Cuál de los siguientes esquemas es más eficiente de los que puedan resolver el problema correctamente?

   a) Esquema voraz.
   b) Esquema de programación dinámica.
   c) Esquema de vuelta atrás.
   d) Esquema de ramificación y poda.

ALGORITMO PLANIFICACIÓN

SO 2018

1. Un dentista pretende dar servicio a *n* pacientes y conoce el tiempo requerido por cada uno de ellos, siendo t_i , $i= 1,2,\ldots,n$ el tiempo requerido por el paciente *i*. El objetivo es minimizar el tiempo medio de estancia de los pacientes en la consulta. En relación a este problema, ¿cuál de las siguientes afirmaciones es **falsa**?

    (a) El esquema más eficiente para resolver este problema correctamente es el esquema voraz.
    (b) El coste del algoritmo voraz que resuelve el problema es O(n log n).
    (c) Suponiendo tres pacientes con tiempos de servicio t_1=15, t_2=60 y t_3=30, el tiempo mínimo de estancia total posible es de tres horas.
    (d) Suponiendo tres pacientes con tiempos de servicio t_1=15, t_2=20 y t_3=30, el tiempo mínimo de estancia total posible es de 115 minutos.

F1 2017 A

1. Un servidor tiene que atender *tres* clientes que llegan todos juntos al sistema. El tiempo que requerirá dar servicio a cada cliente es conocido, siendo t_1=5, t_2=10 y t_3=3. El objetivo es minimizar el tiempo medio de estancia de los clientes en el sistema. Se quiere implementar un algoritmo voraz que construya la secuencia ordenada óptima de servicio a los distintos clientes. Según este algoritmo, el tiempo mínimo de estancia en el sistema del conjunto de clientes es:

    (a) 26
    (b) 29
    (c) 31
    (d) 34

F2 2017 A

1. Un peluquero pretende dar servicio a *n* clientes y conoce el tiempo requerido por cada uno de ellos, siendo t_i, i= 1,2,...,n el tiempo requerido por el cliente *i*. El objetivo es minimizar el tiempo total que todos los clientes están en el sistema, y como el número de clientes es fijo, minimizar la espera total equivale a minimizar la espera media. ¿Cuál de los siguientes esquemas es **más eficiente** de los que puedan resolver el problema correctamente?

    (a) Esquema voraz.
    (b) Esquema de programación dinámica.
    (c) Esquema de vuelta atrás.
    (d) Esquema de ramificación y poda.

F1 2016 A

6. Sea un procesador que ha de atender n procesos. Se conoce de antemano el tiempo que necesita cada proceso. Se desea determinar en qué orden se han de atender los procesos para minimizar la suma del tiempo que los procesos permanecen en el sistema. En relación a este problema, ¿cuál de las siguientes afirmaciones es **cierta**?

   (a) El coste del algoritmo voraz que resuelve el problema es, en el mejor de los casos, $O(n^2)$.
   (b) Suponiendo tres clientes con tiempos de servicio t_1=5, t_2=10 y t_3=3, el tiempo mínimo de estancia posible es de 26 segundos.
   (c) Suponiendo tres clientes con tiempos de servicio t_1=5, t_2=10 y t_3=3, el tiempo mínimo de estancia posible es de 31 segundos.
   (d) Todas las afirmaciones anteriores son falsas.

PROBLEMA MOCHILA

F2 2018 A

1. Sea el problema de la mochila, en el que tenemos una mochila de capacidad *M*, y *n* objetos con beneficios $b_1, b_2, b_3, \ldots b_n$ y pesos $p_1, p_2, p_3, \ldots p_n$. El objetivo es maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta *M*. Indica, de los esquemas siguientes, cuál es el más adecuado en el caso de que cada objeto pueda meterse en la mochila entero o fraccionado.

   (a) El esquema voraz.
   (b) El esquema divide y vencerás.
   (c) El esquema de vuelta atrás.
   (d) El esquema de ramificación y poda.

F2 2017 A

2. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado mediante programación dinámica. Suponga que se dispone de 3 objetos con pesos {9, 6, 5} y que aportan unos beneficios de {38,40,24}, respectivamente. Suponga también que dispone de una mochila con una capacidad máxima de 15. Considere que, al aplicar el algoritmo, los objetos se consideran en el orden dado en el enunciado ({9,6, 5}). Indique qué afirmación es **cierta**:

   (a) El beneficio máximo final obtenido es 80.
   (b) La tabla de resultados parciales en la fila correspondiente al objeto de peso 6 es: 0 0 0 0 0 0 24 24 24 40 40 40 64 64 64 78
   (c) La tabla de resultados parciales en la fila correspondiente al objeto de peso 5 es: 0 0 0 0 0 24 40 40 40 40 40 64 64 64 64 78
   (d) El problema propuesto se resuelve más eficientemente utilizando un enfoque voraz.

SR 2017

2. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado con programación dinámica. Supongamos que se dispone de 5 objetos con pesos: 1,3,4,5,7 y beneficios: 2,5,10,14,15 respectivamente, y un volumen máximo de 8. Identifica cuál de las siguientes respuestas correspondería al contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 5, si dichos objetos se consideran en el orden indicado.

   a. 0 2 2 5 7 7 7 7 7
   b. 0 2 2 5 10 12 12 15 15
   c. 0 2 2 5 10 12 14 16 19
   d. 0 2 2 5 10 14 16 16 19

F1 2016 A

4. Sea el problema de la mochila, en el que tenemos una mochila de capacidad *M*, *n* objetos con beneficios $b_1, b_2, b_3, \dots b_n$ y pesos $p_1, p_2, p_3, \dots p_n$. El objetivo es maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta *M*. Indica, de los esquemas siguientes, cuál es el más adecuado y eficiente en el caso de que cada objeto puede meterse en la mochila entero o fraccionado.

   (a) El esquema voraz utilizando como criterio de selección escoger el objeto de más valor de los que quedan.
   (b) El esquema voraz utilizando como criterio de selección escoger el objeto cuyo valor por unidad de peso sea el mayor de los que quedan.
   (c) El esquema de vuelta atrás calculando todas las soluciones posibles y escogiendo la mejor.
   (d) El esquema de ramificación y poda.

SR 2016

6. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado mediante programación dinámica. Suponga que se dispone de 5 objetos con volúmenes {1,2,5,6,7} y que aportan unos beneficios de {1,6,18,22,28}, respectivamente. Suponga también que dispone de una mochila con una capacidad máxima de 11. Indique cuál sería el contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 7, si dichos objetos se consideran en orden creciente de pesos.

   (a) 0 1 6 7 7 18 19 24 25 25 28 29
   (b) 0 1 6 7 7 18 22 24 28 29 29 40
   (c) 0 1 6 7 7 18 22 28 29 34 35 40
   (d) Ninguna de las anteriores.

SO 2014

3. Sea el problema de la mochila, en el que tenemos una mochila de capacidad *M*, *n* objetos con beneficios $b_1, b_2, b_3, \ldots b_n$ y pesos $p_1, p_2, p_3, \ldots p_n$. El objetivo es maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta *M*. Indica de los esquemas siguientes **cuál es el más adecuado** en el caso de que cada objeto puede meterse en la mochila, no meterse, o meter la mitad del objeto obteniendo la mitad del beneficio.

   (a) El esquema voraz.
   (b) El esquema divide y vencerás.
   (c) El esquema de vuelta atrás.
   (d) El esquema de ramificación y poda.

SO 2013

3. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado mediante programación dinámica. Suponga que se dispone de 5 objetos con volúmenes {1,2,5,6,7} y que aportan unos beneficios de {1,6,18,22,28}, respectivamente. Suponga también que dispone de una mochila con una capacidad máxima de 11. Indique cuál sería el contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 7, si dichos objetos se consideran en orden creciente de pesos.

   a) 0 1 6 7 7 18 19 24 25 25 28 29
   b) 0 1 6 7 7 18 22 24 28 29 29 40
   c) 0 1 6 7 7 18 22 28 29 34 35 40
   d) Ninguna de las anteriores.

SR 2013

2. Sea el problema de la mochila, en el que tenemos una mochila de capacidad *M*, *n* objetos con beneficios $b_1, b_2, b_3, \ldots b_n$ y pesos $p_1, p_2, p_3, \ldots p_n$. El objetivo es maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta *M*. Indique de los esquemas siguientes cuál es el más adecuado en el caso de que cada objeto puede meterse en la mochila entero o fraccionado.

   a) El esquema voraz.
   b) El esquema divide y vencerás.
   c) El esquema de vuelta atrás.
   d) El esquema de ramificación y poda.

F1 2012 A

2.- Sea el problema de la mochila en su versión de objetos no fraccionables solucionado con programación dinámica. Supongamos que se dispone de 5 objetos con pesos: 1,3,4,5,7 y beneficios: 2,5,10,14,15 respectivamente, y un

volumen máximo de 8.Identifica cuál de las siguientes respuestas correspondería al contenido de la tabla de resultados parciales en la fila correspondiente al objeto de peso 5, si dichos objetos se consideran en orden creciente de pesos.

a. 0 2 2 5 10 12 12 15 15
b. 0 2 2 5 10 14 16 16 19
c. 0 2 2 5 10 12 14 16 19
d. Ninguna de las anteriores.

4.- En un problema de mochila con objetos fraccionables tenemos n = 8 objetos disponibles. Los pesos de los objetos son w= (8, 7, 6, 5, 4, 3, 2, 1) y los beneficios son v= (10, 9, 8, 7, 6, 5, 4, 3). ¿Cuál es el beneficio óptimo para este ejemplo, suponiendo que la capacidad de la mochila es M = 9?

a. 12

b. 15
c. 16.5
d. Ninguna de las otras respuestas es correcta.

F2 2012 A

3. Dado el problema de la mochila donde tenemos una mochila de capacidad *M*, *n* objetos con beneficios $b_1, b_2, b_3, \ldots b_n$ y pesos $p_1, p_2, p_3, \ldots p_n$ y siendo el objetivo maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta *M*, en el caso de que los objetos sean fraccionables, la resolución del mismo:

   a. Debe resolverse mediante un esquema voraz ya que no es eficiente realizarlo mediante ramificación y poda.
   b. Debe hacerse mediante ramificación y poda, ya que no es posible resolverlo utilizando un esquema voraz.
   c. Sólo puede resolverse mediante un esquema de programación dinámica.
   d. Puede hacerse mediante un esquema voraz pero no es posible resolverlo mediante ramificación y poda.

ALGORITMO VECTOR

F1 2018 A

3. Se dispone de un vector, V, que almacena números enteros en orden estrictamente creciente, y se desea averiguar si existe algún elemento que cumpla V[i]=i. ¿Cuál sería la estrategia más adecuada para resolver el problema?

   (a) Algoritmo voraz.
   (b) Divide y vencerás.
   (c) Vuelta atrás
   (d) Ramificación y poda

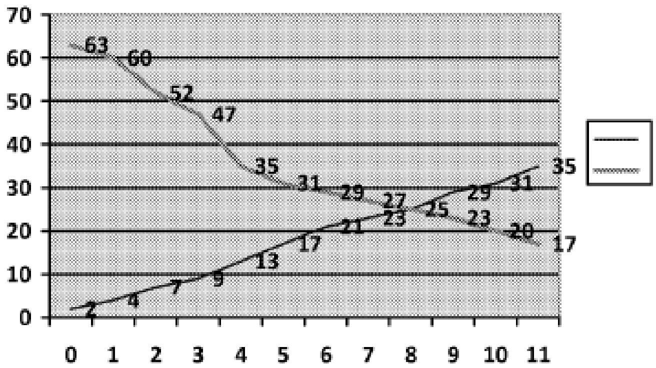
SO 2018

2. Se dispone de un vector, V, que almacena n números naturales, y se desea averiguar si existe algún elemento que aparezca al menos n/2 + 1 veces en el vector. ¿Cuál sería la estrategia más adecuada para resolver el problema?

   (a) Algoritmo voraz.
   (b) Divide y vencerás.
   (c) Vuelta atrás
   (d) Ramificación y poda

F1 2014 A

3. Considere dos vectores *f* y *g* de *n* elementos que representan los valores que toman dos funciones en el intervalo *[0..n-1]*. Los dos vectores están ordenados, pero el primero *f* es un vector estrictamente creciente (*f[0] < f[1] < … < f[n-1]*), mientras *g* es un vector estrictamente decreciente (*g[0] > g[1]>…> g[n-1]*). Las curvas que representan dichos vectores se cruzan en un punto concreto, y lo que se desea saber es si dicho punto está contenido entre las componentes de ambos vectores, es decir, si existe un valor *i* tal que *f[i] = g[i]* para $0 \leq i \leq n - 1$.
   La figura muestra un ejemplo en el que las gráficas se cruzan en *i=8* que se corresponde con el valor 25 en ambas curvas.



   Se busca un algoritmo que compruebe si el punto de cruce está contenido en las componentes de ambos vectores. ¿Cuál de las siguientes opciones es <u>**cierta**</u>?

   (a) Se puede encontrar un algoritmo recursivo de coste logarítmico.
   (b) El algoritmo más eficiente que se puede encontrar es O(n).
   (c) El algoritmo más eficiente que se puede encontrar es $O(n^2)$
   (d) Ninguna de las anteriores.

SR 2014

2. Se tiene un vector V de números enteros no repetidos y ordenados de menor a mayor. Se desea comprobar si existe algún elemento del vector que coincida con su índice (V[i]=i). Indica cuál de las siguientes afirmaciones es **cierta**:

   (a) La solución más eficiente se obtiene mediante programación y dinámica y tiene un coste de O(n).
   (b) Es posible resolver el problema en tiempo logarítmico mediante divide y vencerás.
   (c) La solución más eficiente tiene un coste $O(n^2)$.
   (d) Ninguna de las anteriores es cierta.

F1 2012 A

6.- Se dispone de un vector, V, que almacena números enteros en orden estrictamente creciente, y se desea averiguar si existe algún elemento que cumpla V[i]=i. ¿Cuál sería la estrategia más eficiente para resolver el problema?

a. Algoritmo voraz.
b. Divide y vencerás.
c. Programación dinámica.
d. Ninguna de las anteriores es aplicable al problema

MATRICES PD

SO 2018

6. Dadas las matrices: *A1 (3x5), A2 (5x2), A3 (2x3)* y *A4 (3x2)*,y siendo *E(i,j), i≤j*, el número de operaciones mínimo para resolver la operación $A_i \times A_{i+1} \times \ldots \times A_j$ mediante programación dinámica, se pide indicar cuál de las siguientes opciones es **cierta**:

   (a) E(2,3) = 15
   (b) E(1,3) = 30
   (c) E(2,4) = 32
   (d) E(2,2) = 10

F2 2017 A

3. Dadas las siguientes matrices (cuyas dimensiones se especifican entre paréntesis): A_1 (3x5), A_2 (5x2), A_3 (2x3) y A_4 (3x2) y siendo E(i,j) el número de operaciones mínimo para resolver la operación $A_i \times A_{i+1} \times \ldots \times A_j$ mediante programación dinámica, se pide indicar cuál de las siguientes opciones es **cierta**.

   (a) E(2,3) = 15
   (b) E(1,3) = 30
   (c) E(2,4) = 32
   (d) E(2,2) = 10

F1 2015 A

4. Dadas las matrices: A_1 (3x5), A_2 (5x2) y A_3 (2x3) y A_4 (3x2) y siendo E(i,j) el número de operaciones mínimo para resolver la operación A_i x A_{i+1} x … x A_j mediante programación dinámica, se pide indicar cuál de las siguientes opciones es **cierta**.

(a) E(2,3) = 15
(b) E(1,3) = 30
(c) E(2,4) = 32
(d) E(2,2) = 10

F2 2015 A

2. Dadas las matrices: A_1 (2x2), A_2 (2x5) y A_3 (5x3) y A_4 (3x1) siendo E(i,j) el número de operaciones mínimo para resolver la operación A_i x A_{i+1} x … x A_j mediante programación dinámica, se pide indicar cuál de las siguientes opciones es cierta.

(a) E(2,4) = 25
(b) E(3,4) = 10
(c) E(1,2) = 40
(d) E(2,2) = 10

TIPO ALGORITMO

F1 2018 A

1. Un dentista pretende dar servicio a *n* pacientes y conoce el tiempo requerido por cada uno de ellos, siendo t_i , $i= 1,2,\ldots,n$, el tiempo requerido por el paciente *i*. El objetivo es minimizar el tiempo total que todos los clientes están en el sistema, y como el número de pacientes es fijo, minimizar la espera total equivale a minimizar la espera media. ¿Cuál de los siguientes esquemas es más eficiente de los que puedan resolver el problema correctamente?

   (a) Esquema voraz.
   (b) Esquema de programación dinámica.
   (c) Esquema de vuelta atrás.
   (d) Esquema de ramificación y poda.

F2 2018 A

4. Una filmoteca ha organizado un maratón de cortometrajes. Durante 24 horas se proyectarán cortos de cine (todos diferentes) en las n salas disponibles. Un cinéfilo ha conseguido la programación completa donde aparecen todas las películas que se van a proyectar durante el maratón, incluyendo el título, duración del corto, sala en la que se proyecta y hora de comienzo. Si se quiere planificar el maratón del cinéfilo de forma que pueda ver el máximo número posible de cortos, ¿Cuál es el esquema más apropiado para hacer la planificación eficientemente?

   (a) Esquema voraz.
   (b) Divide y vencerás.
   (c) Esquema de vuelta atrás.
   (d) Esquema de ramificación y poda.

SO 2017

6. Una filmoteca ha organizado un maratón de cine de terror. Durante 24 horas se proyectarán películas (todas diferentes) en las n salas disponibles. Un aficionado a este género de películas ha conseguido la programación completa donde aparecen todas las películas que se van a proyectar durante el maratón. Junto con el título, nombre del director, duración de la película y otros datos de interés, se indica la sala de proyección y la hora de comienzo. Indique qué tipo de algoritmo de entre los siguientes sería el más eficiente para el aficionado planifique su maratón de cine, teniendo en cuenta que su único objetivo es ver el máximo número de posible de películas:

   a. Esquema voraz.
   b. Esquema divide y vencerás
   c. Esquema de vuelta atrás.
   d. Esquema de ramificación y poda.

2. Indica cuál de las siguientes afirmaciones es **falsa**:

a. En un algoritmo de vuelta atrás es posible retroceder para deshacer una decisión ya tomada.
b. La programación dinámica es aplicable a muchos problemas de optimización cuando se cumple el Principio de Optimalidad de Bellman.
c. El problema de calcular el camino de coste mínimo entre cada par de nodos de un grafo (es decir, desde todos los nodos hasta todos los restantes nodos) se resuelve utilizando como base el algoritmo de Dijkstra. La resolución de este problema completo tiene una complejidad de $O(n^2)$.
d. La programación dinámica es apropiada para resolver problemas que pueden descomponerse en subproblemas más sencillos en los que haya llamadas repetidas en la secuencia de llamadas recursivas.

F2 2016 A

2. Indique cuál de las siguientes afirmaciones es **falsa**:

(a) La programación dinámica es aplicable a muchos problemas de optimización cuando se cumple el Principio de Optimalidad de Bellman.
(b) En un algoritmo de vuelta atrás es posible retroceder para deshacer una decisión ya tomada.
(c) El problema de calcular el camino de coste mínimo entre cada par de nodos de un grafo (es decir, desde todos los nodos hasta todos los restantes nodos) se resuelve utilizando el algoritmo de Dijkstra con una complejidad de $O(N^3)$.
(d) La programación dinámica no es apropiada para resolver problemas que pueden descomponerse en subproblemas más sencillos en los que haya llamadas repetidas en la secuencia de llamadas recursivas.

4. Sea el problema de la mochila, en el que tenemos una mochila de capacidad M, n objetos con beneficios $b_1, b_2, b_3, \dots b_n$ y pesos $p_1, p_2, p_3, \dots p_n$. El objetivo es maximizar el valor de los objetos transportados, respetando la limitación de la capacidad impuesta M. Indica, de los esquemas siguientes, cuál es el más adecuado y eficiente en el caso de que cada objeto puede meterse en la mochila entero o fraccionado.

(a) El esquema voraz utilizando como criterio de selección escoger el objeto de más valor de los que quedan.
(b) El esquema voraz utilizando como criterio de selección escoger el objeto cuyo valor por unidad de peso sea el mayor de los que quedan.
(c) El esquema de vuelta atrás calculando todas las soluciones posibles y escogiendo la mejor.
(d) El esquema de ramificación y poda.

5. Para resolver determinado problema hemos diseñado un algoritmo de tipo divide y vencerás. ¿Cuál de las siguientes afirmaciones es **falsa**?:

(a) La resolución debe alcanzar un caso trivial que se pueda resolver sin realizar nuevas descomposiciones.
(b) El problema se resuelve por divisiones sucesivas en subproblemas, que pueden ser de mayor o de menor tamaño que el de partida.
(c) Los algoritmos basados en este esquema pueden requerir un paso de combinación de las soluciones parciales.
(d) El algoritmo de la búsqueda binaria aplica la estrategia divide y vencerás.

SR 2016

1. Indique cuál de las siguientes afirmaciones es **falsa**:

(a) La estrategia divide y vencerás es una técnica algorítmica que se basa en la descomposición de un problema en subproblemas del mismo tipo, lo que permite disminuir la complejidad y en algunos casos paralelizar la resolución de los mismos.
(b) El esquema de vuelta atrás realiza un recorrido en anchura del grafo implícito del espacio de soluciones de un problema, podando aquellas ramas para las que el algoritmo puede comprobar que no pueden alcanzar una solución al problema.
(c) El problema de encontrar todos los ciclos Hamiltonianos de un grafo resuelto con el esquema vuelta atrás tiene una cota del coste de O(n!).
(d) La programación dinámica es apropiada para resolver problemas que pueden descomponerse en subproblemas más sencillos en los que haya llamadas repetidas en la secuencia de llamadas recursivas.

F1 2015 A

3. Se dispone de un conjunto *A* de *n* números enteros (tanto positivos como negativos) sin repeticiones almacenados en una lista. Dados dos valores enteros *m* y C, siendo $m < n$ se desea resolver el problema de encontrar un subconjunto de *A* compuesto por exactamente *m* elementos y tal que la suma de los valores de esos m elementos sea *C*. ¿Cuál de los siguientes esquemas es más eficiente de los que puedan resolver el problema correctamente?

(a) Esquema voraz.
(b) Esquema de divide y vencerás.
(c) Esquema de vuelta atrás.
(d) Esquema de ramificación y poda.

F2 2015

1. Dos socios que conforman una sociedad comercial deciden disolverla. Cada uno de los *n* activos que hay que repartir tiene un valor entero positivo. Los socios quieren repartir dichos activos a medias y, para ello, primero quieren comprobar si el conjunto de activos se puede dividir en dos subconjuntos disjuntos, de forma que cada uno de ellos tenga el mismo valor. ¿Cuál de los siguientes esquemas de los que puedan resolver el problema correctamente es más eficiente?

(a) Esquema voraz.
(b) Esquema de divide y vencerás.
(c) Esquema de vuelta atrás.
(d) Esquema de ramificación y poda.

SR 2015

1. Indique cuál de las siguientes afirmaciones es **cierta**:

   (a) En el problema de la mochila con solución voraz, necesariamente los objetos pueden fraccionarse, de lo contrario no es posible resolverlo mediante un esquema voraz.
   (b) El coste de un algoritmo de ordenación por fusión es lineal si el vector está ya previamente ordenado.
   (c) El mejor método para resolver los problemas de planificación con plazo fijo es mediante el esquema de Vuelta Atrás.
   (d) Ninguna de las anteriores es correcta.

3. En una fontanería se necesitan hacer n reparaciones urgentes y se sabe de antemano el tiempo que va a llevar cada una de ellas: la tarea i-ésima tardará t_i minutos. Es un objetivo de la empresa que los clientes estén satisfechos, y esto se consigue minimizando el tiempo medio de espera de los clientes. Si llamamos E_i a lo que espera el cliente i-ésimo hasta que se repara su avería, se trata de minimizar la expresión:

   $$E(n) = \sum_{i=1}^{n} E_i$$

   La empresa sólo dispone de un fontanero para realizar las reparaciones y quiere minimizar el tiempo medio de espera de los clientes. ¿Cuál de los siguientes esquemas es **más eficiente** de los que puedan resolver el problema correctamente?

   (a) Esquema voraz.
   (b) Esquema de programación dinámica.
   (c) Esquema de vuelta atrás.
   (d) Esquema de ramificación y poda.

F2 2014 A

3. El enemigo ha desembarcado en nuestras costas invadiendo n ciudades. Los servicios de inteligencia han detectado del número de efectivos enemigos en cada ciudad, e_i. Para contraatacar, se dispone de n equipos listos para intervenir y hay que distribuirlos entre las n ciudades. Cada uno de estos equipos consta de d_j efectivos entrenados y equipados. Para garantizar el éxito de la intervención en una ciudad es necesario que contemos al menos con tantos efectivos de defensa como el enemigo.

   Se busca un algoritmo que indique qué equipo debe intervenir en cada ciudad, de forma que se maximice el número de éxitos garantizados. Indica cuál de las siguientes afirmaciones es **cierta**:

   (a) El esquema de ramificación y poda es el único que puede resolver de forma óptima el problema.
   (b) Se puede encontrar una estrategia voraz que permita resolver el problema.
   (c) El esquema de programación dinámica es el único que puede resolver de forma óptima el problema.
   (d) El esquema de vuelta atrás es el más apropiado para resolver este problema.

6. Indique cuál de las siguientes afirmaciones es **cierta**:

   (a) El enfoque de vuelta atrás realiza un recorrido en anchura del árbol implícito que representa el espacio de posibles soluciones del problema.
   (b) Si se utiliza el enfoque de programación dinámica para calcular el camino de coste mínimo entre cada par de nodos de un grafo, la complejidad temporal es de $O(n^2)$.
   (c) Cuando ambos son aplicables, el enfoque de ramificación y poda se comporta de manera más eficiente que el enfoque voraz en la resolución de problemas de optimización.
   (d) El hecho de utilizar un algoritmo voraz para obtener la solución de un problema no garantiza que la solución obtenida sea la óptima.

SO 2014

1. Una filmoteca ha organizado un maratón de cortometrajes. Durante 24 horas se proyectarán cortos de cine (todos diferentes) en las n salas disponibles. Un cinéfilo ha conseguido la programación completa donde aparecen todas las películas que se van a proyectar durante el maratón, incluyendo el título, duración del corto, sala en la que se proyecta y hora de comienzo.
   Si se quiere planificar el maratón del cinéfilo de forma que pueda ver el máximo número posible de cortos, ¿Cuál es el **esquema más apropiado** para hacer la planificación eficientemente?

   (a) Esquema voraz.
   (b) Divide y vencerás.
   (c) Esquema de vuelta atrás.
   (d) Esquema de ramificación y poda.

F1 2013 A

3. Se dispone de n cubos numerados del 1 al n. Cada cubo tiene impresa una letra distinta en cada una de sus caras, aunque distintos cubos pueden tener letras repetidas. Se pretende, dada una palabra de longitud n, colocar el total de cubos de manera consecutiva de forma que se forme la palabra dada. ¿Cuál de los siguientes esquemas algorítmicos es el más apropiado para resolver dicho problema?

   a) Programación dinámica.
   b) Divide y vencerás.
   c) Ramificación y poda.
   d) Vuelta atrás.

6. Considere un array A[1..n] ordenado y formado por enteros diferentes, algunos de los cuales pueden ser negativos. Se busca un algoritmo recursivo que calcule en tiempo logarítmico un índice i tal que $1 \leq i \leq n$ y $T[i] = i$, siempre que este índice exista. Se supone que las operaciones elementales tienen coste unitario. ¿A cuál de los siguientes esquemas correspondería dicho algoritmo?

   a) Esquema voraz.
   b) Divide y vencerás.
   c) Esquema de vuelta atrás.
   d) Esquema de ramificación y poda.

F2 2013 A

1. Indica cuál de las siguientes afirmaciones es correcta:

   a) En el problema de la mochila con solución voraz, necesariamente los objetos pueden fraccionarse, de lo contrario no es posible resolverlo mediante un esquema voraz.
   b) El coste de un algoritmo de ordenación por fusión es lineal si el vector está ya previamente ordenado.
   c) El mejor método para resolver los problemas de planificación con plazo fijo es mediante el esquema de Vuelta Atrás.
   d) Ninguna de las anteriores es correcta.

3. Dos piratas se pretenden repartir a partes iguales el tesoro acumulado en sus años de saqueos. Cada objeto robado tiene un valor. Como ambos piratas tienen sus preferencias por algunos objetos, antes de decidir los que se quedan quieren saber todas las posibles formas de repartir el tesoro en dos partes iguales. Indica qué esquema de los siguientes sería el más adecuado para diseñar un algoritmo que hiciera esa tarea.

   a) Algoritmo voraz.
   b) Vuelta atrás.
   c) Ramificación y poda.
   d) Divide y vencerás.

SR 2013

3. Indique cuál de las siguientes afirmaciones es **falsa**:

   a) La programación dinámica es aplicable a problemas de optimización en los que se cumple el Principio de Optimalidad de Bellman.
   b) En un algoritmo de vuelta atrás es posible retroceder para deshacer una decisión ya tomada.
   c) El problema de calcular el camino de coste mínimo entre cada par de nodos de un grafo (es decir, desde todos los nodos hasta todos los restantes nodos) se resuelve utilizando el algoritmo de Dijkstra con una complejidad de $O(N^2)$.
   d) La programación dinámica es apropiada para resolver problemas que pueden descomponerse en subproblemas más sencillos y cuyas soluciones parciales se solapan.

F2 2012 A

2. En una carrera de coches por el desierto uno de los principales problemas es el abastecimiento de gasolina. Uno de los participantes está decidido a ganar y ha calculado que con el tanque de gasolina lleno puede recorrer M km. sin parar. El participante dispone de un mapa que le indica las distancias entre las gasolineras que hay en la ruta y cree que, parándose a repostar el menor número de veces posible, podrá ganar. Para ayudarle hay que diseñar un algoritmo que le sugiera en qué gasolineras debe hacerlo. Hay que tener en cuenta que hay una única ruta posible. Indique de los esquemas siguientes cuál es el más adecuado:

   a. El esquema voraz.
   b. El esquema programación dinámica.
   c. El esquema de vuelta atrás.
   d. El esquema de ramificación y poda.

SO 2012

2. Los residentes de una ciudad no quieren pavimentar todas sus calles (que son de doble sentido), sino sólo aquellas que les permitan ir de una intersección a otra cualquiera de la ciudad con comodidad. Quieren gastarse lo menos posible en la pavimentación, teniendo en cuenta que el coste es directamente proporcional a la longitud de las calles que hay que pavimentar. El alcalde querría saber qué calles tiene que pavimentar para gastarse lo menos posible. ¿Cuál de los siguientes esquemas es más eficiente de los que puedan resolver el problema correctamente?

   a. Esquema voraz.
   b. Esquema de programación dinámica.
   c. Esquema de vuelta atrás.
   d. Esquema de ramificación y poda.

5. Indique cuál de las siguientes afirmaciones es cierta:

   a. La programación dinámica se puede emplear en muchos de los problemas que se resuelven utilizando el esquema divide y vencerás.
   b. La programación dinámica tiene como principal ventaja que supone un decremento tanto en coste computacional como en espacio de almacenamiento.
   c. El esquema de programación voraz se aplica a problemas de optimización en los que la solución se puede construir paso a paso comprobando, en cada paso, la secuencia de decisiones tomadas anteriormente.
   d. El esquema de ramificación y poda es el más indicado a utilizar en el caso que se deseen encontrar todas las soluciones posibles a un problema dado.

SR 2102

4. Se desea codificar en forma binaria un texto expresado mediante un conjunto de caracteres. Por tanto, cada carácter vendrá codificado en forma de una cadena de bits única. Por ejemplo, la 'a' se codificará con su cadena de bits correspondiente siempre que aparezca en el texto. Se conocen los datos referentes a la frecuencia con la que aparecen cada uno de los caracteres que forman parte del texto que se desea codificar. Puesto que el objetivo de la codificación es minimizar el número de bits a escribir, se debe asignar a aquel carácter más frecuente la cadena de bits más corta, siempre que no haya sido ya asignada, y así de forma sucesiva para el resto de caracteres dependiendo de su frecuencia. ¿Qué esquema algorítmico de resolución sería el más adecuado?

   a. Voraz.
   b. Vuelta Atrás.
   c. Divide y Vencerás.
   d. Ramificación y Poda.

OTROS

SR 2018

1. Elije la afirmación **correcta** sobre el esquema de Ramificación y Poda :

   (a) Los nodos pendientes de visitar siguen un orden basado en el principio "primero en llegar, último en salir".
   (b) Los nodos pendientes de visitar siguen un orden basado en el principio "primero en llegar, primero en salir".
   (c) El orden de exploración se reorganiza en cada paso del algoritmo, ya que se basa en un montículo.
   (d) Ninguna de las otras es cierta.

3. Sea un tablero de ajedrez de dimensión n x n, y un caballo situado en cierta posición de salida (i,j). Se desea conocer la secuencia de movimientos que permiten al caballo recorrer todas las casillas del tablero sin repetir ninguna. ¿Cuál de los siguientes esquemas es más apropiado de los que puedan resolver el problema correctamente?

   (a) Esquema voraz.
   (b) Esquema divide y vencerás .
   (c) Esquema de vuelta atrás.
   (d) Esquema de ramificación y poda.

6. Indicar la respuesta **cierta** acerca del coste de los algoritmos del esquema de Divide y Vencerás

   (a) El coste que se consigue es siempre igual o más eficiente que O (log n)
   (b) El coste que se consigue es siempre igual o más eficiente que O (n log n).
   (c) El coste no puede ser en ningún caso peor que el cuadrático O (n^2).
   (d) Ninguna de las otras es cierta.

F1 2017 A

6.- Se dispone de un vector, V, que almacena números enteros en orden estrictamente creciente, y se desea averiguar si existe algún elemento que cumpla V[i]=i. ¿Cuál sería la estrategia más eficiente para resolver el problema?

a. Algoritmo voraz.
b. Divide y vencerás.
c. Programación dinámica.
d. Ninguna de las anteriores es aplicable al problema

F2 2017 A

6. ¿Cuál de las siguientes afirmaciones es **falsa** con respecto al coste de las funciones de manipulación de grafos?

   (a) La función `AñadirVertice` que añade un vértice a un grafo y devuelve el grafo con dicho vértice incluido, tiene un coste constante cuando el grafo se implementa con una matriz de adyacencia.
   (b) La función `BorrarArista` es menos costosa cuando el grafo se implementa mediante una matriz de adyacencia.

   (c) La operación `Adyacente?`, que comprueba si dos nodos son adyacentes, es más costosa cuando el grafo se implementa con una lista de adyacencia.
   (d) La función `Adyacentes`, que devuelve una lista con los vértices adyacentes a uno dado, es más costosa cuando el grafo se implementa con una matriz de adyacencia.

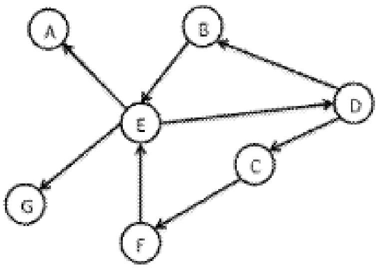
SR 2017

4. Elije la afirmación correcta sobre el esquema de Ramificación y Poda:

   a. Los nodos pendientes de visitar se almacenan en una cola con comportamiento "primero en llegar, primero en salir".
   b. Los nodos pendientes de visitar se almacenan en una pila con comportamiento "primero en llegar, último en salir".
   c. Los nodos pendientes de visitar se almacenan en un montículo y de esta forma, el recorrido es siempre en anchura.
   d. Ninguna de las otras es cierta.

F1 2016 A

1. Sea el grafo dirigido de la siguiente figura:



¿Cuál de las siguientes afirmaciones es **cierta**?

   (a) Un posible recorrido en anchura visitaría los nodos en orden {D,B,C,E,F,A,G}, si se toma el nodo D como nodo de partida.
   (b) Un posible recorrido en profundidad visitaría los nodos en orden {D,B,E,A,G,C,F}, si se toma el nodo D como nodo de partida.
   (c) El nodo E es un punto de articulación.
   (d) Todas las afirmaciones anteriores son ciertas.

2. Indica cuál de las siguientes afirmaciones es **cierta**:

(a) El coste del algoritmo para fusionar dos subvectores ordenados que se utiliza en la ordenación por fusión es $O(n)$, siendo n la suma de los tamaños de los dos vectores.
(b) El coste del algoritmo que soluciona el problema del coloreado de grafos mediante el esquema de vuelta atrás es $O(n^m)$, siendo m el número de colores y n el número de nodos.
(c) El coste del algoritmo de Kruskal es $O(n^2)$, siendo n el número de nodos del grafo.
(d) El coste del algoritmo de ordenación rápida en el caso mejor es $O(n^2)$, siendo n el número de elementos del array.

5. Para resolver determinado problema hemos diseñado un algoritmo de tipo divide y vencerás. ¿Cuál de las siguientes afirmaciones es **falsa**?:

(a) La resolución debe alcanzar un caso trivial que se pueda resolver sin realizar nuevas descomposiciones.
(b) El problema se resuelve por divisiones sucesivas en subproblemas, que pueden ser de mayor o de menor tamaño que el de partida.
(c) Los algoritmos basados en este esquema pueden requerir un paso de combinación de las soluciones parciales.
(d) El algoritmo de la búsqueda binaria aplica la estrategia divide y vencerás.

F2 2016 A

1. Dado un grafo con *n* vértices y *a* aristas sobre el que se pueden realizar las siguientes operaciones:

- *esAdyacente*: comprueba si dos vértices son adyacentes.
- *borrarVertice*: borra el vértice especificado y todas sus aristas.
- *añadirVertice*: añade un nuevo vértice al grafo.

¿Cuál es la complejidad de cada una de las operaciones anteriores según se utilice una matriz de adyacencia (*MA*) o una lista de adyacencia (*LA*)?

(a)

	MA	LA
esAdyacente	O(1)	O(1)
borrarVértice	O(n)	O(n)
añadirVértice	O(n)	O(1)

(b)

	MA	LA
esAdyacente	O(1)	O(n)
borrarVértice	O(1)	O(n+a)
añadirVértice	O(1)	O(1)

(c)

	MA	LA
esAdyacente	O(1)	O(n)
borrarVértice	O(n)	O(n+a)
añadirVértice	O(n)	O(1)

(d)

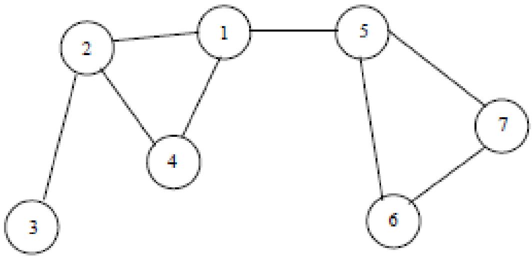
	MA	LA
esAdyacente	O(1)	O(1)
borrarVértice	O(n)	O(n)
añadirVértice	O(n)	O(n)

6. En relación al esquema algorítmico de vuelta atrás, ¿cuál de las siguientes afirmaciones es **cierta**?:

(a) Implementa un recorrido en amplitud de forma recursiva sobre el grafo implícito del problema.
(b) Aun existiendo una solución al problema, puede darse el caso de que el esquema de vuelta atrás no la encuentre.
(c) Siempre es más eficiente que el esquema voraz, de forma que, si ambos son aplicables, nos decantaremos por utilizar el esquema de vuelta atrás.
(d) Todas las afirmaciones anteriores son falsas.

SO 2016

5. Dado el grafo de la figura:



Los valores, al finalizar el algoritmo de cálculo de los puntos de articulación son:

(a) numOrden[]=[1,2,3,4,5,6,7] y bajo[]=[1,1,3,1,5,5,5]
(b) numOrden[]=[1,2,3,4,5,6,7] y bajo[]=[1,2,3,4,5,6,5]
(c) numOrden[]=[1,3,7,3,2,5,6] y bajo[]=[1,2,3,4,5,6,5]
(d) Ninguno de los anteriores

6. Dado el siguiente algoritmo:

```
// Precondición: i pertenece a {1,2,3}
hanoi(n,i,j) {
     si n=1 entonces escribe "Mover de "  i   "hasta"   j
     sino {
          hanoi(n-1, i    , 6-i-j)
          hanoi(1   , i    , j)
          hanoi(n-1, 6-i-j, j)
     }
}
```

El coste asintótico temporal pertenece al orden:

(a) O(2n)
(b) $O(2^n)$
(c) $O(\log_2 n)$
(d) $O(n^2)$

F1 2015 A

6. ¿Cuál de las siguientes afirmaciones es **falsa** con respecto al coste de las funciones de manipulación de grafos?

   (a) La función `Etiqueta` que devuelve la etiqueta o peso asociado a la arista que une dos vértices tiene un coste constante cuando el grafo se implementa con una matriz de adyacencia.
   (b) La función `BorrarArista` es más costosa cuando el grafo se implementa mediante una lista de adyacencia.
   (c) La operación `Adyacente?`, que comprueba si dos nodos son adyacentes, es más costosa cuando el grafo se implementa con una matriz de adyacencia .
   (d) La función `Adyacentes`, que devuelve una lista con los vértices adyacentes a uno dado, es menos costosa cuando el grafo se implementa con una matriz de adyacencia.

F2 2105 A

5. El problema de la liga de equipos consiste en que dados *n* equipos con $n = 2^k$, se desea realizar una liga de forma que cada equipo puede jugar un partido al día y la liga debe celebrarse en *n*-1 días, suponiendo que existen suficientes campos de juego. El objetivo sería realizar un calendario que indique el día que deben jugar cada par de equipos. Si este problema se resuelve con un esquema divide y vencerás, considerando que el caso trivial se da cuando la liga consta de 2 equipos, la descomposición se realiza dividiendo el problema en dos partes similares, y la combinación se produce dando los subproblemas por resueltos y aplicando el principio de inducción. Indica de entre los siguientes, cuál sería el coste mínimo de dicho algoritmo.

   (a) $\Theta(n)$.
   (b) $\Theta(n^2)$.
   (c) $\Theta(n \log n)$.
   (d) $\Theta(n^2 \log n)$.

SO 2015

1. ¿Cuál de las siguientes afirmaciones es **cierta** con respecto al coste de algunos algoritmos y su eficiencia?

   (a) El algoritmo de Kruskal tiene un coste que está en $O(n \log n)$.
   (b) El algoritmo de Prim cuando se combina el uso de la lista de adyacencia con el uso de un montículo es más eficiente cuando el grafo es denso porque de esta manera el coste está en $O(n^2 \log n)$.
   (c) El coste del algoritmo Quicksort en el caso peor es de orden $O(n \log n)$.
   (d) La función que crea un montículo a partir de una colección de valores, si se utiliza el procedimiento Hundir puede llegar a tener un coste lineal.

4. Durante la ejecución de un algoritmo de Ramificación y Poda se halla una solución que es mejor que la mejor solución existente en ese momento y que mejora la estimación optimista de la cima del montículo. Esto implica que:

   (a) Definitivamente el algoritmo ha encontrado la solución.
   (b) Se actualiza la cota y se sigue con la exploración porque no hemos terminado.
   (c) Se actualiza la cima del montículo y se sigue con la exploración porque no hemos terminado.
   (d) Ninguna de las anteriores es correcta.

SR 2015

5. Unos hermanos gemelos quieren repartirse los regalos de su fiesta de cumpleaños que no indican a cuál de los dos van destinados. Sus padres han asignado a cada regalo un valor. Como ambos niños tienen sus preferencias por algunos regalos, antes de decidir los que se quedan quieren saber todas las posibles formas de repartir los regalos en dos partes de igual valor. Indique qué esquema de los siguientes sería el **más adecuado** para diseñar un algoritmo que hiciera esa tarea.

   (a) Algoritmo voraz.
   (b) Vuelta atrás.
   (c) Ramificación y poda.
   (d) Divide y vencerás.

F1 2014 A

4. ¿Cuál de las siguientes afirmaciones es **falsa**?

   (a) El algoritmo de ordenación por fusión (*mergesort*) es O(n log n).
   (b) La eficiencia del algoritmo de ordenación rápida (*quicksort*) es independiente de que el pivote sea el elemento de menor valor del vector.
   (c) El algoritmo de ordenación rápida en el caso peor es $O(n^2)$.
   (d) El algoritmo de ordenación basada en montículos (*heapsort*) es O(n log n).

F2 2014 A

1. Indique cuál de las siguientes afirmaciones es **cierta**:

   (a) El esquema de divide y vencerás obtiene siempre soluciones eficientes de coste O(log n) o O(n·log n)
   (b) La búsqueda en profundidad recursiva obtiene siempre una eficiencia logarítmica
   (c) La ordenación por Quicksort tiene coste $O(n^2)$ en el caso peor
   (d) Ninguna de las anteriores es cierta.

4. Indica de entre los siguientes, cuál sería el coste mínimo de un algoritmo que dado un vector C[1..n] de números enteros distintos no ordenado, y un entero S, determine si existen o no dos elementos de C tales que su suma sea exactamente S.

   (a) $\Theta(n)$.
   (b) $\Theta(n^2)$.
   (c) $\Theta(n \log n)$.
   (d) $\Theta(n^2 \log n)$.

SR 2014

5. Cuál de las siguientes afirmaciones **no es cierta** en relación con la implementación de los grafos mediante matriz y mediante listas de adyacencia:

   (a) Añadir un vértice es O(1) en la implementación mediante listas de adyacencia.
   (b) Borrar un vértice es de orden O(n) usando matriz de adyacencia.
   (c) Comprobar si un vértice es adyacente a otro tiene complejidad O(n) usando matriz de adyacencia.
   (d) Añadir una arista tiene complejidad O(1) en ambas implementaciones.

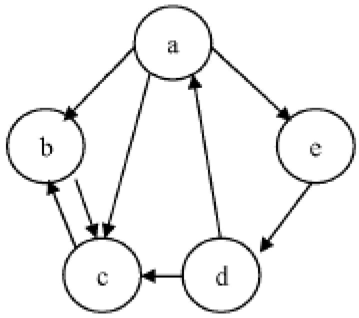
F1 2013 A

2. Durante la ejecución de un algoritmo de Ramificación y Poda se halla una solución que es mejor que la mejor solución existente en ese momento y que mejora la estimación optimista de la cima del montículo. Esto implica:

   a) Definitivamente el algoritmo ha encontrado la solución.
   b) Se actualiza la cota y se sigue con la exploración porque no hemos terminado.
   c) Se actualiza la cima del montículo y se sigue con la exploración porque no hemos terminado.
   d) Ninguna de las anteriores es correcta.

F2 2013 A

5. ¿Cuántos componentes fuertemente conexos existen en el grafo de la siguiente figura?



a) Uno solo, formado por todo el grafo.
b) Dos.
c) Tres.
d) Más de tres.

SO 2013

2. En el problema de colorear un grafo con n nodos utilizando m colores, de manera que no haya dos vértices adyacentes que tengan el mismo color, una cota superior ajustada del coste de encontrar una solución utilizando un esquema adecuado es del orden de:

a) $O(n^2)$.
b) $O(m^n)$.
c) $O(m \log n)$.
d) $O(n^m)$.

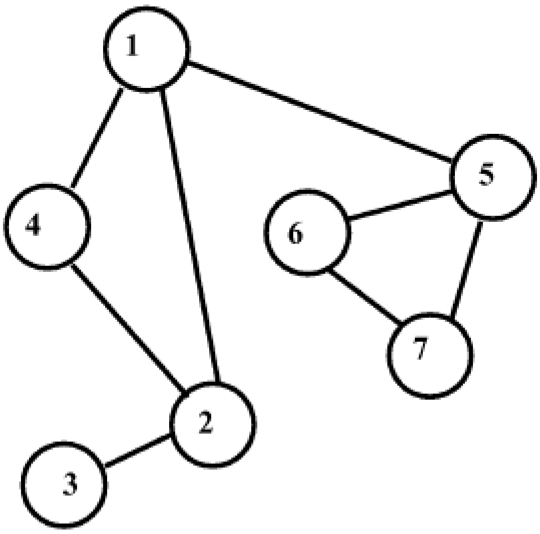
SR 2013

4. Se tiene que resolver un problema de tipo divide y vencerás de tamaño n donde el decrecimiento del problema es en progresión aritmética a razón de dos llamadas recursivas con tamaño n-1, siendo el resto de las operaciones realizadas en el algoritmo de coste O(k) con k constante. El coste del algoritmo es:

a) $O(n^2)$
b) $O(2^n)$
c) $O(n^2 \cdot \log n)$
d) $O(2n)$

F1 2012 A

1. Dado el grafo no dirigido de la figura, la aplicación del algoritmo de cálculo de los puntos de articulación (y de su árbol de recubrimiento asociado) da como resultado el vector *bajo[]* siguiente:



a.- [1,1,3,1,5,5,5]
b.- [1,2,3,4,1,6,7]
c.- [1,3,3,1,1,3,3]
d.- Ninguna de las anteriores

F2 2012 A

1. Identifique cuál de las siguientes secuencias de código implementa un recorrido en anchura de una componente conexa de un grafo:

   a.

```
fun RecAnchura(v: nodo, visitado: Vector)
var
   u,w: nodo
   Q: TCola
fvar
Q ← ColaVacía
Encolar(v,Q)
mientras ¬ vacia(Q) hacer
    visitado[v] ← cierto
    u ← Primero(Q)
    Desencolar(u,Q)
    para cada w adyacente a u hacer
      si ¬ visitado[w]  entonces
         visitado[w] ← cierto
      fsi
    Encolar(w,Q)
    fpara
mientras
ffun
```

   b.

```
fun RecAnchura(v: nodo, visitado: Vector)
var
   u,w: nodo
   Q: TCola
fvar
Q ← ColaVacía
visitado[v] ← cierto
Encolar(v,Q)
mientras ¬ vacia(Q) hacer
    u ← Primero(Q)
    Desencolar(u,Q)
    para cada w adyacente a u hacer
      si ¬ visitado[w]  entonces
         visitado[w] ← cierto
         Encolar(w,Q)
```

```
        fsi
      fpara
  mientras
  ffun
```

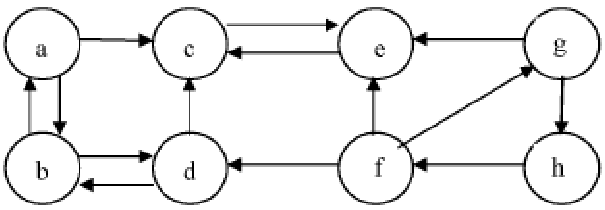
c.

```
  fun RecAnchura(v: nodo, visitado: Vector)
  var
     u,w: nodo
     Q: TCola
  fvar
  Q ← ColaVacía
  visitado[v] ← cierto
  Encolar(v,Q)
  u ← Primero(Q)
  mientras ¬ vacia(Q) hacer
      Desencolar(u,Q)
      para cada w adyacente a u hacer
        si ¬ visitado[w]  entonces
             Encolar(w,Q)
        fsi
       visitado[w] ← cierto
    fpara
  mientras
  ffun
```

d.

Ninguna de las secuencias anteriores es correcta.

3. ¿Cuántos componentes fuertemente conexos existen en el grafo de la siguiente figura?



a. Uno solo, formado por todo el grafo.
b. Dos.
c. Tres.
d. Más de tres.

SO 2012

3. Suponga un algoritmo que dadas dos listas de tamaños *n* y *m*: $A = a_1, \ldots a_n$ y $B = b_1, \ldots, b_m$ dé como resultado la lista de aquellos elementos que pertenecen a *A* pero no a *B*. Se asume que *A* y *B* no contienen elementos duplicados, pero no que éstos estén acotados. Indique cuál de los siguientes costes se ajusta más al del algoritmo que puede resolver el problema con menor coste:

    a. O(n log m + m log m).
    b. O(n log m + m^2).
    c. Máx (O (n log n), O(m log m)).
    d. O(nm).

4. Dado un grafo con *n* vértices y *a* aristas sobre el que se pueden realizar las siguientes operaciones:

    - *esAdyacente*: comprueba si dos vértices son adyacentes.
    - *borrarVértice*: Borra el vértice especificado y todas sus aristas.
    - *añadirVértice*: Añade un nuevo vértice al grafo.

    ¿Cuál es la complejidad de cada una de las operaciones anteriores según se utilice una matriz de adyacencia (*MA*) o una lista de adyacencia (*LA*)?

a)

	MA	**LA**
esAdyacente	O(1)	O(1)
borrarVértice	O(n)	O(n)
añadirVértice	O(n)	O(1)

b)

	MA	**LA**
esAdyacente	O(1)	O(n)
borrarVértice	O(1)	O(n+a)
añadirVértice	O(1)	O(1)

c)

	MA	**LA**
esAdyacente	O(1)	O(n)
borrarVértice	O(n)	O(n+a)
añadirVértice	O(n)	O(1)

d)

	MA	**LA**
esAdyacente	O(1)	O(1)
borrarVértice	O(n)	O(n)
añadirVértice	O(n)	O(n)

SR 2012

1. Dado el siguiente algoritmo:

```
función X(int n)
        si n>0 entonces
                escribir "n";
                X(n-1);
                escribir "n-1";
        fsi
```

   Indica cuál de las siguientes sería la salida al ejecutar la llamada “X(5)”:
   a. 5 4 3 2 1 0 4 3 2 1 0.
   b. 5 4 3 2 1 0 1 2 3 4.
   c. 5 4 3 2 1 0 1 2 3 4 0.
   d. 5 4 3 2 1 0 4 3 2 1.