Grafos

FEBRERO 2015 – SEMANA 1

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.

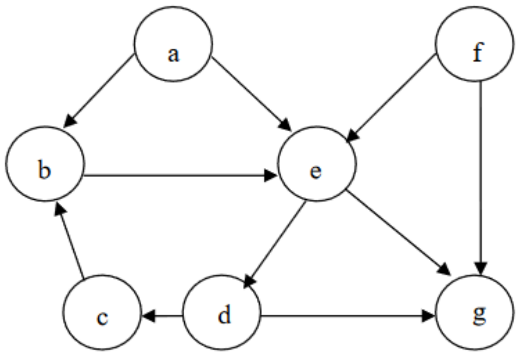
(b) La función BorrarArista es más costosa cuando el grafo se implementa mediante una lista de adyacencia.

(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.

FEBRERO 2014 – _SEMANA 2 (MODELO A)

2. ¿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.

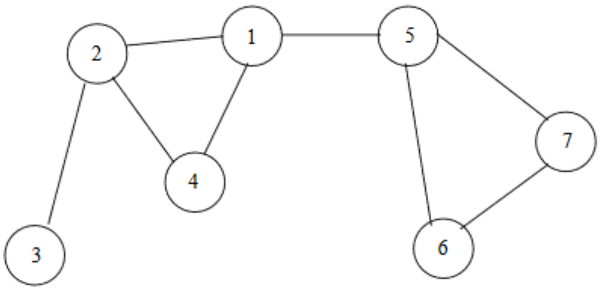
FEBRERO 2014 – _SEMANA 1 (MODELO A)

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

(a) El grado de un vértice de un grafo no dirigido es el número de aristas que salen o entran en él.
(b) Un camino en un grafo dirigido es una secuencia finita de arcos entre dos vértices, tal que el vértice del extremo final de cada arco coincide con el del extremo inicial del arco siguiente.
(c) Un grafo nulo es un grafo sin vértices.
(d) La longitud de un camino es el número de aristas y nodos que contiene.

SEPTIEMBRE 2016 – _SEMANA 1 (MODELO A)

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

FEBRERO 2016 – _SEMANA 2 (MODELO 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)

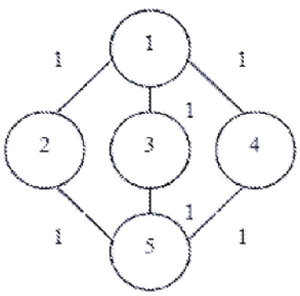
(d)

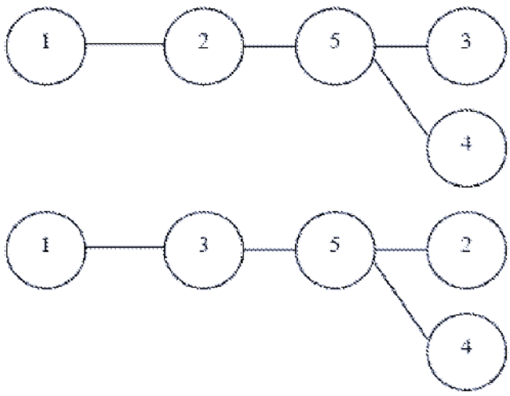
	MA	LA
esAdyacente	O(1)	O(1)
borrarVértice	O(n)	O(n)
añadirVértice	O(n)	O(n)

FEBRERO 2018 – _SEMANA 1 (MODELO A)

6. Selecciona la afirmación más ajustada de las siguientes. Las siguientes tres figuras corresponden a:

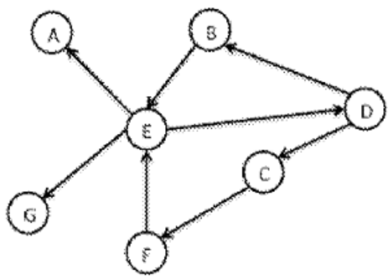
   (a) Las tres corresponden a tres grafos no dirigidos sin ninguna posible relación entre ellos.
   (b) La de la izquierda es un grafo y a los grafos no dirigidos conexos con aristas de igual coste no se les puede asociar más de un árbol de recubrimiento mínimo.
   (c) La de la izquierda es un grafo y las de la derecha son dos posibles árboles de recubrimiento mínimo asociados a él.
   (d) Ninguna de las anteriores es correcta.





FEBRERO 2016 – _SEMANA 1 (MODELO 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.

FEBRERO 2020 – SEMANA 2 (MODELO A)

3. Con respecto al recorrido en amplitud o en anchura de un grafo indica cuál de las siguientes afirmaciones es **falsa**:

   (a) El coste del recorrido en anchura es $O(n^2)$ cuando se representa mediante una matriz de adyacencia y $O(n+a)$ si se representa con listas de adyacencia.
   (b) Se emplea para realizar exploraciones parciales de un grafo potencialmente infinito.
   (c) El recorrido en anchura es de naturaleza recursiva y la estructura de datos que se corresponde con el recorrido es de tipo pila.
   (d) Se puede considerar un recorrido por niveles del grafo ya que dado un nodo inicial primero se visitan los nodos que están a una arista de distancia.

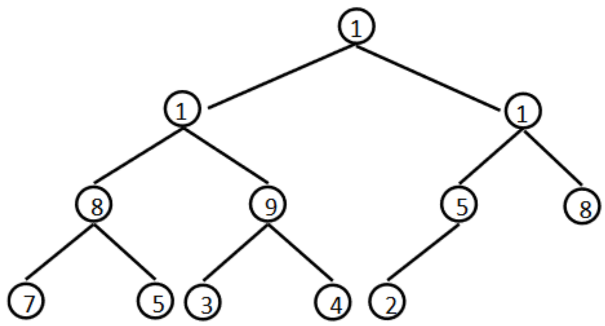
5. Respecto a la estructura de datos grafo, ¿Cuál de las siguientes afirmaciones es **falsa**?

   (a) El máximo número de aristas en un grafo dirigido de n vértices es n(n-1).
   (b) Un recorrido en profundidad de un grafo es equivalente al recorrido en postorden de un árbol.
   (c) Si el grafo no es conexo, el recorrido en profundidad le asocia un bosque de árboles, uno por cada componente conexa del árbol.
   (d) En un grafo representado mediante una matriz de adyacencia el coste de la búsqueda en profundidad es de $O(n^2)$.

Montículos

FEBRERO 2014 – _SEMANA 1 (MODELO A)
ANULADA

6. Dado el siguiente montículo:

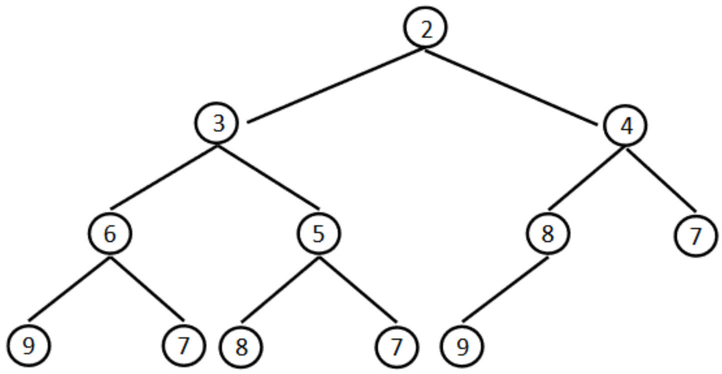


Indique cuál de las siguientes afirmaciones es **falsa**:

(a) El montículo propuesto es un montículo de máximos.
(b) El vector que lo representa de forma correcta es [15,13,12,8,9,5,8,7,5,3,4,2].
(c) El orden de complejidad de la operación de inserción de un nuevo elemento en el montículo es O(n).
(d) Los montículos son estructuras de datos de gran utilidad a la hora de implementar colas de prioridad.

SEPTIEMBRE 2016 – _ORIGINAL (MODELO A)

1. Dado el siguiente montículo:



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

(a) El montículo propuesto es un montículo de mínimos.
(b) El vector que lo representa de forma correcta es [2,3,4,6,5,8,7,9,7,8,7,9].
(c) La operación “mínimo” en un montículo binomial tiene un coste O(1).
(d) El orden de complejidad de la operación de borrado de un elemento en el montículo es O(log n).

SEPTIEMBRE 2015 – _ORIGINAL (MODELO A)

6. Considérese el vector [10,6,3,5,2,3,2] que representa un montículo. ¿Cuál sería la representación resultante de insertar (función *Insertar* del texto base) en este montículo el valor 6 usando la función flotar?

(a) [10,6,6,3,3,2,5,2]
(b) [10,6,5,3,6,2,2,3]
(c) [10,6,3,6,2,3,2,5]
(d) Ninguna de las opciones anteriores.

FEBRERO 2018 – _SEMANA 1 (MODELO A)

2. Respecto a la estructura de datos montículo de mínimos, cuál de las siguientes afirmaciones es **falsa**:

(a) Con un montículo de mínimos disponemos de una estructura de datos en la que encontrar el mínimo es una operación de coste constante.
(b) El montículo sirve de apoyo a la creación de un algoritmo de ordenación eficiente conocido como Heapsort.
(c) El montículo es un árbol binario que puede estar o no balanceado.
(d) Cuando se extrae el primer elemento de un montículo, restaurar la propiedad de montículo tiene coste O(log n).

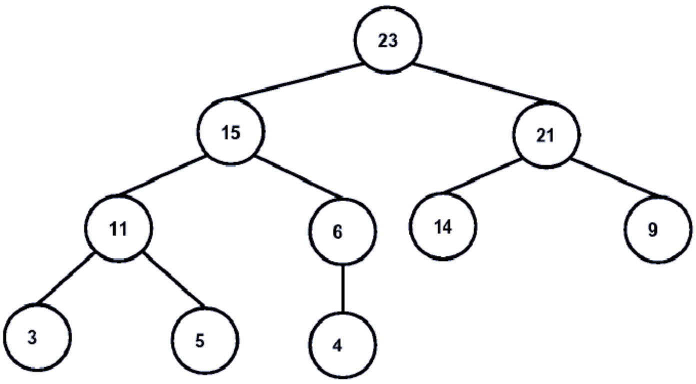
FEBRERO 2018 – _SEMANA 2 (MODELO A)

6. Considérese el vector [15,7,10,5,3,8,2] que representa un montículo de máximos. ¿Cuál sería la representación resultante de insertar en este montículo el valor 11 usando la función flotar?

(a) [15,11,10,7,5,8,2,3]
(b) [15,11,10,7,3,8,2,5]
(c) [15,10,11,7,3,8,2,5]
(d) Ninguna de las opciones anteriores.

FEBRERO 2017 – _SEMANA 2 (MODELO A)

4. Dado el siguiente montículo de máximos, ¿Cuál de los siguientes vectores lo representa de forma **correcta**?



(a) 23, 15, 21, 11, 6, 14, 9, 3, 5, 4
(b) No es un montículo de máximos, sino de mínimos
(c) 23, 15, 11, 3, 5, 6, 4, 21, 14, 9
(d) 23, 15, 21, 11, 6, 3, 5, 4, 14, 9

SEPTIEMBRE 2017 – _ORIGINAL (MODELO A)

5. Considérese el vector *v=[3,6,4,7,9,2,5]*. ¿Cuál de las siguientes opciones es cierta?

a. El vector *v* es un montículo de mínimos.
b. *v* sería un montículo de mínimos si se flotara el elemento de valor 2.
c. *v* sería un montículo de mínimos si se hundiera el elemento de valor 6.
d. Ninguna de las opciones anteriores.

FEBRERO 2016 – _SEMANA 1 (MODELO A)

3. Considérese el vector v[1..n] = [2,3,5,3,4,5,7,5,6]. Indica cuál de las siguientes opciones es **cierta**:

(a) El vector v es un montículo de mínimos.
(b) El vector v no es un montículo de mínimos porque el elemento v[5] = 4 debe ser flotado.
(c) El vector v no es un montículo de máximos porque el elemento v[5] = 4 debe ser hundido.
(d) Ninguna de las anteriores.

SEPTIEMBRE 2013 – ORIGINAL (MODELO A)

1. Considérese el vector *v=[2,6,8,12,7,5,4]*. ¿Cuál de las siguientes opciones es cierta?

a) El vector *v* es un montículo de mínimos.
b) *v* sería un montículo de mínimos si se flotara el elemento de valor 5.
c) *v* sería un montículo de mínimos si se hundiera el elemento de valor 8.
d) Ninguna de las opciones anteriores.

FEBRERO 2020 – SEMANA 1 (MODELO A)

1. Considérese el vector v[1..n] = [8,6,5,5,4,3,2,2] cuál de las siguientes opciones es **cierta**:

(a) El vector v es un montículo de máximos.
(b) El vector v no es un montículo de máximos porque el elemento v[5] = 4 debe ser flotado.
(c) El vector v no es un montículo de máximos porque el elemento v[4] = 5 debe ser hundido.
(d) Ninguna de las anteriores.

FEBRERO 2020 – SEMANA 2 (MODELO A)

1. En relación a los montículos, indique cuál de las siguientes afirmaciones es **falsa**:

(a) El vector [10, 6, 6, 3, 3, 2, 5, 2] es un montículo de máximos.
(b) Al insertar el valor 6 (función *Insertar* del texto base) utilizando la función *flotar* en el montículo [10, 6, 3, 5, 2, 3, 2], la representación resultante es [10,6, 3, 6, 2, 3, 2, 5].
(c) Sea el montículo [6, 5, 4, 4, 1, 3, 2], al obtener la cima del mismo (función *ObtenerCima* del texto base) y restaurar la propiedad de montículo, la representación resultante es [5, 4, 4, 2, 1, 3].
(d) Todas las anteriores son falsas.

Tablas Hash

FEBRERO 2014 – SEMANA 1 (MODELO A)

1. Se tiene una tabla hash con n= 23 índices ocupados y de tamaño M= 1000. El factor de carga **δ** será:

   (a) 0,023, es decir n/M
   (b) 0,046, es decir 2n/M
   (c) -3,7723, es decir Ln (n/M)
   (d) Ninguna de las anteriores

SEPTIEMBRE 2016 – SEMANA 1 (MODELO A)

2. Indica cuál de las siguientes afirmaciones relativas a las tablas de dispersión es **falsa**:

   (a) Una función hash debe repartir los valores h(x) en la tabla de manera equiprobable.
   (b) Una función hash debe ser eficiente e indeterminista.
   (c) En la resolución de colisiones, el recorrido basado en una expresión cuadrática en lugar de lineal permite una mayor dispersión de las colisiones.
   (d) Cambios pequeños en la clave deben resultar en cambios significativos en la función hash h(x).

FEBRERO 2016 – _SEMANA 2 (MODELO A)

3. Respecto a las tablas Hash, podemos afirmar que:

   (a) En el método de hashing cerrado con recorrido mediante doble hashing, las funciones h y h' que se aplican pueden ser iguales cuando el número de elementos de la tabla, m, cumple determinadas condiciones.
   (b) En la resolución de colisiones, el método de hashing abierto es siempre más eficiente que el hashing cerrado.
   (c) En el recorrido mediante doble hashing, la función h'(x) debe cumplir necesariamente que $h'(x) \neq 0$.
   (d) En la resolución de colisiones, si el factor de carga es 1 se puede resolver mediante hashing cerrado.

FEBRERO 2019 – SEMANA 1 (MODELO A)

6. Sea una tabla hash con m=5 y h(x) = x mod 5 usando para la resolución de colisiones un hashing abierto con 3 niveles combinado con encadenamiento directo. En ella se van insertando en orden los siguientes valores: 41,58,16,66,77,12,25,21,31,14. Indique cuál de las siguientes afirmaciones es **cierta**:

   (a) Tras insertar el 25 la tabla hash no ha necesitado realizar encadenamiento.
   (b) Tras insertar el 14 la posición 0 de la tabla carece de valores.
   (c) Tras insertar el valor 77 la tabla hash tiene la posición 1 incompleta.
   (d) Tras insertar el valor 21 la tabla tiene todos los niveles completos.

FEBRERO 2019 – SEMANA 2 (MODELO A)

1. Con respecto a la resolución de colisiones en las funciones Hash se puede afirmar que:

   (a) El recorrido cuadrático es aplicable siempre que se trate de hashing abierto.
   (b) Si el factor de carga es 1 se puede resolver mediante hashing cerrado.
   (c) El hashing abierto contempla un método de resolución conocido como "recorrido mediante doble hashing".
   (d) Ninguna de las anteriores es correcta.

FEBRERO 2018 – SEMANA 1 (MODELO A)

5. Indica cuál de las siguientes afirmaciones es cierta con respecto a la resolución de colisiones:

(a) El método de hashing abierto es siempre más eficiente que el hashing cerrado para la resolución de colisiones.
(b) En el método de hashing cerrado con recorrido lineal existe una probabilidad muy baja de colisiones independientemente de los patrones de las claves.
(c) En el método de hashing cerrado con recorrido mediante doble hashing, la función h y h' que se aplican deben ser iguales cuando el número de elementos de la tabla, m, cumple determinadas condiciones.
(d) El método de hashing cerrado con recorrido cuadrático permite una mayor dispersión de las colisiones por la tabla que el hashing cerrado con recorrido lineal.

FEBRERO 2017 – SEMANA 1 (MODELO A)

3. Se tiene una tabla hash de tamaño m=11 y las funciones $h_1(k)$ = k mod m y $h_2(k)$ = (k mod (m – 1)) +1. Se pide insertar los valores [22,1,13,11,24,33,18,42,31] mediante doble hashing usando $h(k,i) = (h_1(k) + ih_2(k))$ mod m, donde *i* indica el incremento en caso de colisión.

(a)

0	1	2	3	4	5	6	7	8	9	10
22	1	13		11	18	31	24	33	42	

(b)

0	1	2	3	4	5	6	7	8	9	10
22	1	13	18	11		31	24	33	42	

(c)

0	1	2	3	4	5	6	7	8	9	10
22	1	13	11		18	31	24	33	42	

(d)

0	1	2	3	4	5	6	7	8	9	10
22	1	13	24	33	18	31		11	42	

SEPTIEMBRE 2017 – _ORIGINAL (MODELO A)

3. Tenemos una tabla hash de tamaño 7 para almacenar mediante recorrido lineal valores naturales con h(x) = x mod 7. Indicar la respuesta correcta sobre la tabla resultante de insertar en este orden los valores: 0,11,3,7,1,9.

Índice	0	1	2	3	4	5	6
Clave							

a. El índice 3 tiene clave 3
b. El índice 1 tiene clave 1
c. El índice 2 no está ocupado por ningún valor
d. El índice 6 tiene valor 9

SEPTIEMBRE 2013 – _ORIGINAL (MODELO A)

5. Indica cuál de las siguientes afirmaciones es cierta con respecto a la resolución de colisiones:

   a) El método de hashing abierto es siempre más eficiente que el hashing cerrado para la resolución de colisiones.
   b) En el método de hashing cerrado con recorrido lineal existe una probabilidad muy baja de colisiones independientemente de los patrones de las claves.
   c) En el método de hashing cerrado con recorrido cuadrático es posible garantizar que se van a recorrer todos los elementos de la tabla cuando el número de elementos de la tabla, m, cumple determinadas condiciones.
   d) En el método de hashing cerrado con recorrido mediante doble hashing la función h y h' que se aplican pueden ser iguales cuando el número de elementos de la tabla, m, cumple determinadas condiciones.

FEBRERO 2020 – SEMANA 1 (MODELO A)

6. En una tabla de dispersión hash, ¿cuál de las siguientes afirmaciones es **cierta**?

   (a) Dos o más claves diferentes del dominio inicial pueden proporcionar la misma clave en el índice de la tabla hash.
   (b) El uso de tablas hash siempre mejora la eficiencia temporal de los algoritmos.
   (c) Las tablas hash permiten un ahorro considerable de memoria independientemente de la proporción entre los tamaños del dominio inicial y del dominio de claves hash.
   (d) Ninguna de las afirmaciones anteriores es cierta.

Algoritmos Voraces

FEBRERO 2015 – SEMANA 2

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:

(a) {1,6}, {1,7}, {6,2}, {6,3}, {4,6}, {6,5}, {6,8}.

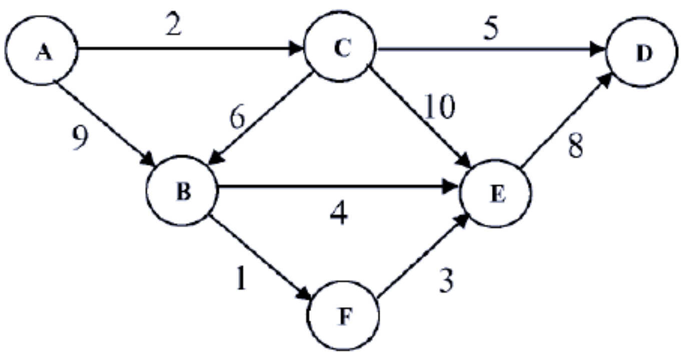
(b) {1,6}, {6,2}, {2,3}, {3,8}, {4,6}, {6,5}, {6,7}.

(c) {1,7}, {1,6}, {6,3}, {6,2}, {4,6}, {6,5}, {6,8}.

(d) {1,6}, {1,2}, {2,3}, {4,6}, {6,5}, {6,8}, {6,7}.

FEBRERO 2015 – SEMANA 1

2. Dado el grafo de la siguiente figura:

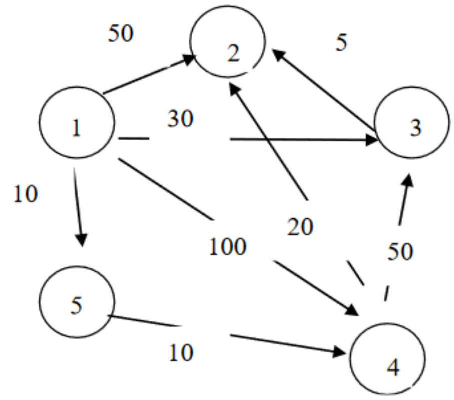


indicad cuál sería el orden en que se seleccionarían (pasan a estar explorados) los nodos al aplicar el algoritmo de Dijkstra desde el nodo A:

(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

FEBRERO 2014 – _SEMANA 1 (MODELO A)

2. Dado el siguiente grafo dirigido:

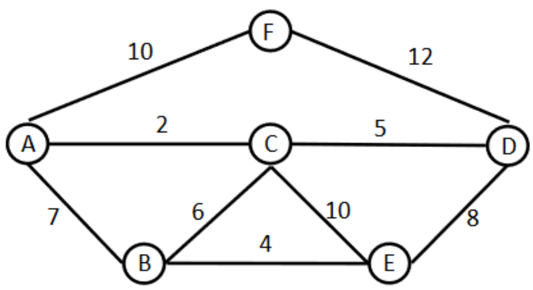


Indique cuál sería el valor del vector ***especial[]*** en el primer y penúltimo paso del algoritmo de Dijkstra:

(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

FEBRERO 2014 – _SEMANA 2 (MODELO 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

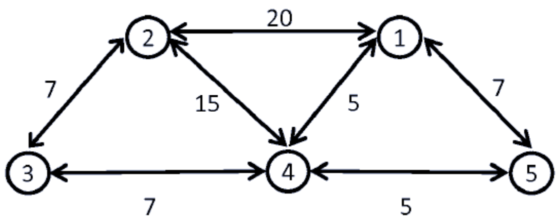
SEPTIEMBRE 2016 – _SEMANA 1 (MODELO A)

4. Para resolver determinado problema hemos diseñado un algoritmo voraz. Indique cuál de las siguientes afirmaciones es **falsa**:

(a) Si encontramos un contraejemplo en el que el algoritmo no alcanza la solución óptima debemos probar con el siguiente valor distinto que nos proporcione la función de selección.
(b) Si encontramos un contraejemplo en el que el algoritmo no alcanza la solución óptima podemos afirmar que no es correcto.
(c) Necesitamos una demostración de optimalidad para poder asegurar que el algoritmo alcanza la solución óptima.
(d) La función de selección escoge al mejor de los candidatos restantes.

FEBRERO 2019 – _SEMANA 1 (MODELO A)

1. Dado el siguiente grafo:



Indique el valor del vector de distancias *especial[]* en el paso del algoritmo de Dijkstra en el que se selecciona el nodo *v=3,* tomando como nodo origen el nodo 1:

(a) [19,12,5,7]
(b) [20,12,5,7]
(c) [20,19,5,7]
(d) Ninguna de las anteriores.

FEBRERO 2019 – _SEMANA 2 (MODELO A)

5. Con respecto al algoritmo de Prim indique cuál de estas afirmaciones es **falsa**:

(a) El algoritmo parte necesariamente de un nodo que hay que seleccionar y que será la raíz del árbol de recubrimiento.
(b) El algoritmo finaliza cuando el árbol de recubrimiento contiene *n-1* aristas, siendo *n* el número de nodos.
(c) Si para representar el grafo se utiliza una lista de adyacencia junto con un montículo para representar los candidatos pendientes el coste del algoritmo es *O(a log n)*, lo que resulta más apropiado que usar una matriz de adyacencia cuando el grafo es denso.
(d) La función de selección en cada paso añade al árbol de recubrimiento una arista de coste mínimo tal que la estructura resultante sigue siendo un árbol.

FEBRERO 2018 – _SEMANA 1 (MODELO 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.

FEBRERO 2018 – _SEMANA 2 (MODELO 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.

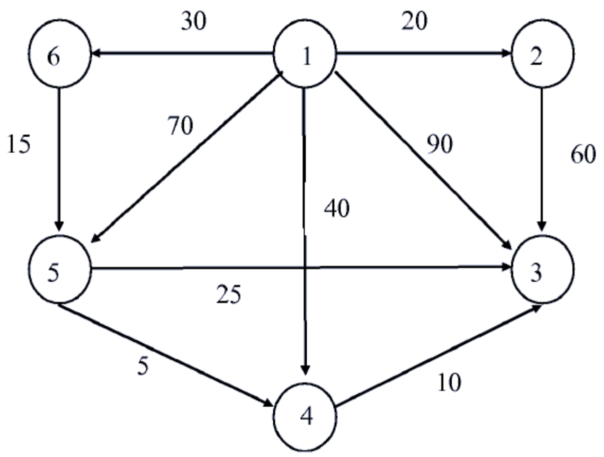
FEBRERO 2017 – _SEMANA 1 (MODELO A)

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

FEBRERO 2017 – _SEMANA 1 (MODELO A)

4. Dado el grafo de la figura:

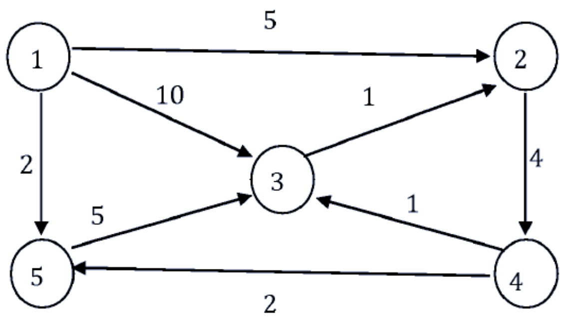


Indique cuál sería el orden en que se seleccionan los nodos del conjunto de candidatos al aplicar el algoritmo de Dijkstra comenzando por el nodo 1:

(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

FEBRERO 2017 – SEMANA 2 (MODELO 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]

SEPTIEMBRE 2017 – ORIGINAL (MODELO A)

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}.

FEBRERO 2020 – SEMANA 2 (MODELO A)

6. Sea el problema de planificación con plazos en el que se han de realizar cuatro trabajos, cada uno de los cuales ha de finalizarse antes de la fecha fi indicada para producir el beneficio b_i. Cada trabajo se realiza en una máquina que consume una unidad de tiempo y solo hay una máquina disponible. Se han de seleccionar los trabajos y la secuencia en la que deben realizarse para que el beneficio total sea máximo:

i	1	2	3	4
f_i	2	1	2	1
b_i	100	10	15	25

Indicar cuál de las siguientes afirmaciones es **cierta**:

(a) La secuencia (2,1,3) es una solución factible.
(b) La solución (1,3) es óptima.
(c) El beneficio máximo alcanzable es 115.
(d) El algoritmo voraz que resuelve este problema considera los trabajos en orden creciente de beneficios siempre que el conjunto de trabajos sea una solución factible.

FEBRERO 2015 – SEMANA 2 (MODELO 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)***

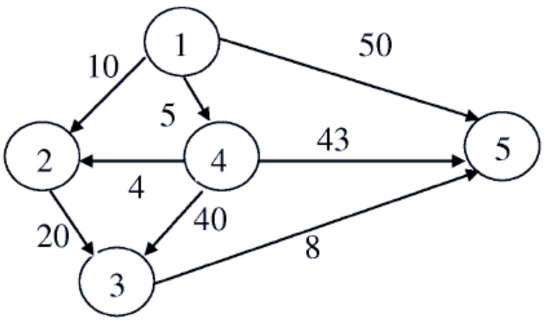
7 → 1(7) ***8 → 3(2), 4(3), 7(2)***

Cuál de las siguientes NO es un valor correcto para el vector "Especial" D[1] 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]

FEBRERO 2020 – SEMANA 1 (MODELO A)

2. 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) 4 2 5 3
(b) 4 3 5 2
(c) 4 2 3 5
(d) 4 3 2 5

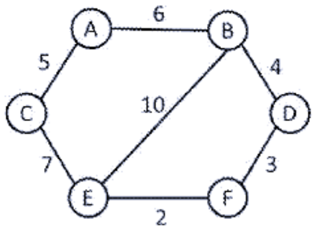
FEBRERO 2020 – SEMANA 1 (MODELO A)

5. Con respecto al algoritmo de Kruskal indica cuál de estas afirmaciones es **falsa**:

(a) En los pasos intermedios del algoritmo el conjunto de aristas del árbol de recubrimiento mínimo que se va formado constará de varias componentes conexas.
(b) El algoritmo finaliza cuando el árbol de recubrimiento contiene *n*-1 aristas, siendo *n* el número de nodos.
(c) El coste de este algoritmo es O(a log n).
(d) La función de selección en cada paso añade al árbol de recubrimiento una arista de coste mínimo no evaluada si une dos nodos pertenecientes a la misma componente conexa.

FEBRERO 2020 – SEMANA 2 (MODELO A)

2. Sea el grafo de la figura:



Indique cuál sería la primera arista que rechaza el algoritmo de Kruskal en la creación del árbol de expansión mínimo:

(a) Ninguna arista.
(b) La arista (E,B).
(c) La arista (C,E).
(d) Ninguna de las anteriores respuestas es válida.

FEBRERO 2018 – _SEMANA 2 (MODELO 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, ... b_n y pesos p_1, p_2, p_3, ... 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.

Respuesta a) Esquema voraz.

SEPTIEMBRE 2017

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.

La c) El camino más corto entre un nodo y los restantes se selecciona con un algoritmo Voraz, tomando como base el algoritmo de Dijkstra, el cual tiene un coste de O(n2). Cuando se da cada par de nodos del grafo dicho coste es O(n3).

FEBRERO 2018 – _SEMANA 2 (MODELO 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.

La a) Esquema voraz.

FEBRERO 2017 – _SEMANA 2 (MODELO 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.

La a) Esquema voraz, con el que se resuelven problemas de minimización de tiempo en el sistema.

FEBRERO 2018 – _SEMANA 1 (MODELO 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,…,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.

La a) Esquema voraz, con el que se resuelven problemas de minimización de tiempo en el sistema.

SEPTIEMBRE 2017 – _ORIGINAL (MODELO A)

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.

La a) Esquema voraz.

FEBRERO 2016 – _SEMANA 1 (MODELO 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.

La 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.

FEBRERO 2020 – SEMANA 1 (MODELO A)

3. Sea el problema de la mochila, en el que tenemos una mochila de capacidad M, n objetos con beneficios $b_1, b_2, b_{3,\dots}\ b_n$ 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** en el caso de que cada objeto puede meterse en la mochila entero, en cualquier fracción, o bien no meterse.

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

Respuesta la a) El esquema voraz.

FEBRERO 2016 – SEMANA 1 (MODELO 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.

Respuesta d) Todas son falsas

FEBRERO 2014 – _SEMANA 2 (MODELO 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.

Respuesta la b) No obstante, que se pueda encontrar no quiere decir que siempre se encuentre y realmente el más apropiado es un divide y vencerás.

FEBRERO 2014 – _SEMANA 2 (MODELO A)

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.

Respuesta d) El hecho de utilizar un algoritmo voraz para obtener la solución deun problema no garantiza que la solución obtenida sea la óptima.

Divide y Vencerás

FEBRERO 2015 – SEMANA 2

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.

FEBRERO 2015 – SEMANA 2

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) $\Theta(n^2 \log n)$.

FEBRERO 2015 – SEMANA 1

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.

FEBRERO 2014 – _SEMANA 2 (MODELO 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.

SEPTIEMBRE 2016 – _SEMANA 2 (MODELO A)

4. 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)

FEBRERO 2019 – _SEMANA 1 (MODELO A)

4. Con respecto al esquema divide y vencerás indique cuál de las siguientes afirmaciones es **verdadera**:

(a) En algunos casos permite la paralelización de la resolución de los subproblemas del mismo tipo.
(b) El coste de este tipo de algoritmos depende del número de subproblemas en los que se descompone el problema pero no de la reducción del tamaño en las sucesivas llamadas recursivas.
(c) Siempre es necesario combinar las soluciones de los subproblemas.
(d) Este esquema aplica el principio de deducción sobre los diversos ejemplares del problema.

FEBRERO 2019 – _SEMANA 1 (MODELO A)

5. Considérese el vector [5,3,7,6,3,9,2,4,8,2]. 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) [2,2,3,4,3] y [9,6,7,8]
(b) [2,3,2,4,3] y [9,6,8,7]
(c) [2,2,3,3,4] y [6,8,7,9]
(d) Ninguna de las opciones anteriores.

SEPTIEMBRE 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.

FEBRERO 2016 – _SEMANA 1 (MODELO A)

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.

FEBRERO 2018 – _SEMANA 2 (MODELO 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.

FEBRERO 2016 – _SEMANA 1 (MODELO A)

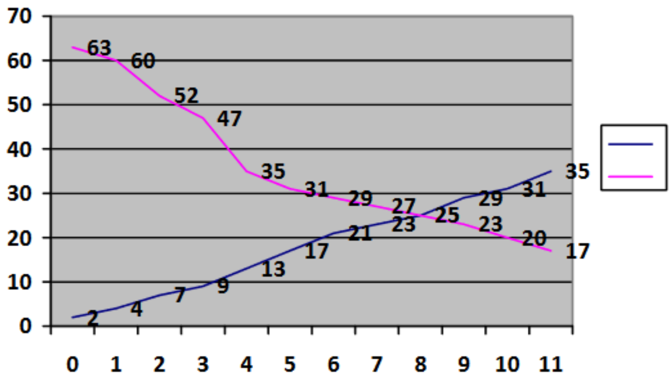
2. Indica cuál de las siguientes afirmaciones es **cierta**:

(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.

La a) El algoritmo es mergesort

FEBRERO 2014 – _SEMANA 1 (MODELO 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 ≤ i ≤ 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 **cierta**?

(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.

Respuesta a) Con el algoritmo Divide y Vencerás, cuyo coste logarítimo es O(log n)

FEBRERO 2014 – _SEMANA 1 (MODELO 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).

Falso b) Si siempre eliges el pivote de menor valor, o si el vector esta ordenado de menor a mayor y siempre coges el primer elemento, obtendrás coste O(n2).

FEBRERO 2018 – SEMANA 1 (MODELO 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

Respuesta b) Divide y vencerás: El problema del que se habla en el enunciado se trata de una búsqueda binaria en un vector ordenado, donde dado un venctor V [1..n] de elementos ordenados se puede verificiar la pertenencia de un valor x para dicho vector mediante un algoritmo de Divide y Vencerás.

FEBRERO 2014 – _SEMANA 2 (MODELO A)

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) $\Theta(n^2 \log n)$.

Respuesta c) Lo que podría hacerse es ordenar el vector, y luego despreciar todos los casos de números donde n > S. Tras ello separamos a la mitad el vector que nos queda, y entonces, en otro vector, almacenar los valores de S menos cada elemento de la primera mitad, y comparar con los elementos de la segunda mitad mediante divide y vencerás, como, por ejemplo, para ver si tengo coincidencia (El resultado aproximado es N log N).

SEPTIEMBRE 2016 – SEMANA 1 (MODELO A)

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) $O(n^2)$

Respuesta b) O(2n)

Programación dinámica

FEBRERO 2015 – SEMANA 1

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

FEBRERO 2015 – SEMANA 2

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 Ai 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

FEBRERO 2019 – _SEMANA 1 (MODELO A)

2. Un lejano país emite *n* sellos diferentes de valores naturales positivos $s_1, s_2, \ldots, s_n$. Se quiere enviar una carta y se sabe que la correspondiente tarifa postal es *T*. Se quiere saber de cuántas formas diferentes se puede franquear exactamente la carta, si el orden de los sellos no importa. ¿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.

FEBRERO 2019 – _SEMANA 2 (MODELO A)

3. Sea el problema de la mochila en su versión de objetos no fraccionables solucionado mediante programación dinámica. Suponga que se dispone de 4 objetos con volúmenes {1,2,5,6} y que aportan unos beneficios de {1,5,15,20}, respectivamente. Suponga también que dispone de una mochila con una capacidad máxima de 12. Indique cuál sería el contenido de la tabla de resultados en la fila correspondiente al objeto de volumen 6, si dichos objetos se consideran en orden creciente de volúmenes.

   (a) 0 1 5 6 6 15 16 20 21 25 26 35 36
   (b) 0 1 5 6 6 15 20 21 25 25 26 35 36
   (c) 0 1 5 6 6 15 21 25 25 26 26 35 36
   (d) Ninguna de las anteriores.

SEPTIEMBRE 2019 (MODELO A)

6. Sean $A_1..A_6$ matrices de dimensiones 30x35, 35x15, 15x5, 5x10, 10x20 y 20x25; y sea m[i,j] el número mínimo de productos escalares para multiplicar $A_i \cdot .. \cdot A_j$; es decir, el subrango *i..j* con $i,j \in \{1..6\}$, siendo *i* la abscisa y *j* la ordenada, y siendo la matriz con los diferentes valores m[i,j] la siguiente:

	1	2	3	4	5	6
6	15,125	10,500	5,375	3,500	5,000	0
5	11,875	7,125	2,500	1,000	0	
4	9,375	4,375	750	0		
3	7,875	2,625	0			
2	15,750	0				
1	0					

¿Cuál de las siguientes opciones es la parametrización óptima del producto de las seis matrices $A_1..A_6$?

(a) $(A_1(A_2A_3)(A_4A_5)A_6)$
(b) $(A_1A_2)(A_3A_4)(A_5A_6)$
(c) $(A_1(A_2(A_3A_4)A_5)A_6)$
(d) Ninguna de las anteriores

FEBRERO 2017 – SEMANA 1 (MODELO 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.

FEBRERO 2019 – _SEMANA 2 (MODELO A)

6. Indique cuál de las siguientes afirmaciones es **falsa** con respecto al esquema de programación dinámica:

(a) El objetivo de este esquema es la reducción del coste del algoritmo mediante la memorización de soluciones parciales que se necesitan para llegar a la solución final.
(b) Es igual de eficiente que el esquema divide y vencerás cuando en este último esquema se dan llamadas recursivas que se repiten en la secuencia de llamadas recursivas.
(c) El problema de la multiplicación asociativa de matrices resuelto con programación dinámica tiene un coste temporal de $O(N^3)$, siendo *N* el número de matrices para multiplicar.
(d) El problema de la distancia de edición entre dos cadenas resuelto con programación dinámica tiene un coste temporal de O(nm), siendo *n* la longitud de una cadena y *m* la longitud de la otra.

FEBRERO 2017 – _SEMANA 2 (MODELO 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.

FEBRERO 2017 – _SEMANA 2 (MODELO 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

SEPTIEMBRE 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.

FEBRERO 2016 – _SEMANA 2 (MODELO 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.

La d)

FEBRERO 2015 – SEMANA 1

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, ... 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, ... 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, ... m^n}, siendo m>1 y n>0.

La cierta es la c.

FEBRERO 2018 – _SEMANA 2 (MODELO 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, …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.

Respuesta la c) Se puede resolver tanto por programación dinámica como por ramificación y poda

Vuelta Atrás

FEBRERO 2016 – _SEMANA 2 (MODELO A)

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.

FEBRERO 2019 – _SEMANA 2 (MODELO A)

4. 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) El esquema de vuelta atrás siempre encontrará una solución al problema, si es que ésta existe.
   (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.

SEPTIEMBRE 2013 – _ORIGINAL (MODELO A)

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) $O(n^m)$.

FEBRERO 2015 – SEMANA 1

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.

Respuesta c) Vuelta atrás

FEBRERO 2015 – SEMANA 2

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. → Su coste es $O(2^n)$

(d) Esquema de ramificación y poda.

Generalmente, todos los ejercicios relativos a de dividir empresas o sociedades se tratan de esquemas de Vuelta Atrás.

Respuesta c) Esquema de vuelta atr.as Su coste es O(2n).

Ramificación y Poda

FEBRERO 2016 – _SEMANA 2 (MODELO A)

4. Se quiere realizar una recopilación de poesías preferidas en un rollo de papel que se puede escribir por ambas caras. Se dispone de una lista de *n* poesías favoritas, junto con la longitud individual de cada una. Lamentablemente, el rollo de *longitud L* no tiene capacidad para contener todas las poesías, por lo que se les ha asignado una puntuación (cuanto más favorita es, mayor es la puntuación). Se pretende obtener el mejor rollo posible según la puntuación, teniendo en cuenta que las poesías deben caber enteras y no es admisible que una poesía se corte al final de una de las caras. ¿Cuál de los siguientes esquemas es más eficiente 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.

FEBRERO 2016 – _SEMANA 2 (MODELO A)

5. Una empresa de montajes tiene n montadores con distintos rendimientos según el tipo de trabajo. Se trata de asignar los próximos *n* encargos, uno a cada montador, minimizando el coste total de todos los montajes. Para ello se conoce de antemano la tabla de costes *C[1..n,1..n]* en la que el valor c_{ij} corresponde al coste de que el montador *i* realice el montaje *j*. ¿Cuál de los siguientes esquemas es más eficiente de los que puedan resolver el problema correctamente?

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

FEBRERO 2019 – _SEMANA 2 (MODELO A)

2. Durante la ejecución de un algoritmo de minimización por Ramificación y Poda, la estimación optimista de la cima del montículo es mayor o igual que la cota superior hasta el momento. Esto implica:

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

FEBRERO 2020 – SEMANA 2 (MODELO A)

4. En relación a los esquemas algorítmicos, ¿cuál de las siguientes afirmaciones es **falsa**?

(a) El esquema de vuelta atrás realiza un recorrido en profundidad del grafo implícito de un problema.
(b) El esquema de ramificación y poda utiliza un montículo para establecer una cola de prioridad de los nodos aún sin explorar.
(c) Preferiremos el esquema de ramificación y poda al esquema voraz siempre que ambos sean aplicables.
(d) El objetivo del esquema de programación dinámica es reducir el coste del algoritmo mediante la memorización de las soluciones parciales.

C) Preferimos el esquema de ramificación y poda al esquema voraz siempre que ambos sean aplicables.

SEPTIEMBRE 2019 (MODELO A)

1. 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 $T = \{m^0, m^1, m^2, \ldots m^n\}$ 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) Ninguna de las anteriores es cierta.

La correcta es la C.

SEPTIEMBRE 2019 (MODELO A)

2. Con respecto al esquema de ramificación y poda, indica cuál de las siguientes afirmaciones es **falsa**:

   (a) En el problema de la distancia de edición entre dos cadenas, si se consideran tres operaciones de transformación, siendo n y m la longitud de las cadenas en caracteres, y n menor o igual que m, una cota superior del espacio de búsqueda generado por el algoritmo de ramificación y poda sería 3^m.
   (b) El montículo se utiliza en este esquema porque permite establecer una cola de prioridad de los nodos que aún no se han explorado.
   (c) Los nodos en el montículo se mantienen ordenados en orden decreciente si la optimización es minimización o creciente si la optimización es maximización del valor de la función objetivo.
   (d) En la fase de poda se eliminan las ramas que no pueden llegar a una solución y las que no pueden mejorar el valor de la función objetivo de la mejor solución alcanzada.

La C es la falsa.