

Normas de valoración del examen:

- La nota del examen representa el 80% de la valoración final de la asignatura (el 20% restante corresponde a las prácticas).
- Cada cuestión contestada correctamente vale 1 punto.
- Cada cuestión contestada incorrectamente baja la nota en 0.3 puntos.
- Debe obtenerse un mínimo de 3 puntos en las cuestiones para que el problema sea valorado (con 3 cuestiones correctas y alguna incorrecta el examen está suspenso).
- La nota total del examen debe ser al menos de 4.5 para aprobar.
- **Las cuestiones se responden en una hoja de lectura óptica.**

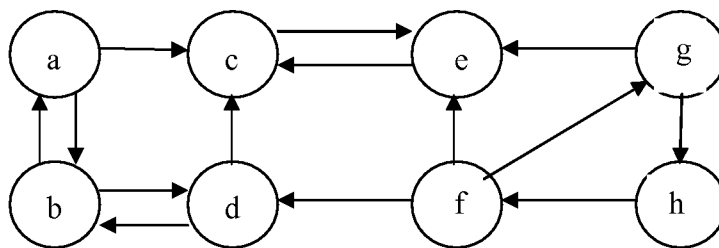
Examen tipo B:

Cuestiones:

1. 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 $[5,6,9,3,4]$

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.
3. Identifique cuál de las siguientes secuencias de código implementa un recorrido en anchura de una componente conexas 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.

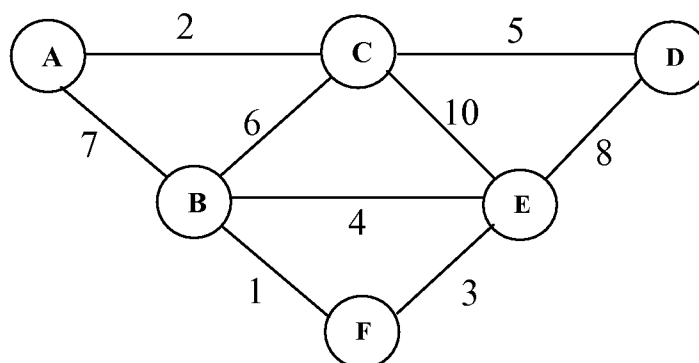
4. En una carrera de coches por el desierto uno de los principales problemas es el abastecimiento de gasolina. Uno de los participantes está decidido a ganar y ha calculado que con el tanque de gasolina lleno puede recorrer M km. sin parar. El participante dispone de un mapa que le indica las distancias entre las gasolineras que hay en la ruta y cree que, parándose a repostar el menor número de veces posible, podrá ganar. Para ayudarlo 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:

- El esquema voraz.
- El esquema programación dinámica.
- El esquema de vuelta atrás.
- El esquema de ramificación y poda.

5. Dado el problema de la mochila donde tenemos una mochila de capacidad M , n objetos con beneficios $b_1, b_2, b_3, \dots, b_n$ y pesos $p_1, p_2, p_3, \dots, p_n$ 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:

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

6. 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:

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

Problema (4 puntos).

Se quiere realizar en un soporte secuencial (cinta de dos caras) una recopilación de canciones preferidas. Se dispone de una lista de n canciones favoritas, junto con la duración individual de cada una. Lamentablemente, la cinta de T minutos no tiene capacidad para contener todas las canciones, por lo que se les ha asignado una puntuación (cuanto más favorita es, mayor es la puntuación). Se pretende obtener la mejor cinta posible según la puntuación, teniendo en cuenta que las canciones deben caber enteras y no es admisible que una canción se corte al final de una de las caras.

La resolución de este problema debe incluir, por este orden:

- Elección del esquema más apropiado, el esquema general y explicación de su aplicación al problema (0,5 puntos).
- Descripción de las estructuras de datos necesarias (0.5 puntos).
- Algoritmo completo a partir del refinamiento del esquema general (2,5 puntos).
- Estudio del coste del algoritmo desarrollado (0.5 puntos).