

Grado en Ingeniería Informática y Grado en Ingeniería en Tecnologías de la Información

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. 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.

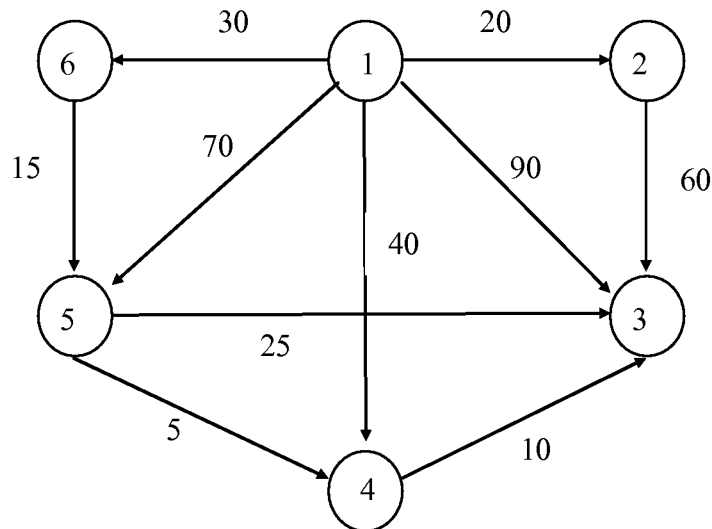
2. En el problema de colorear un grafo con n nodos utilizando m colores, de manera que no haya dos vértices adyacentes que tengan el mismo color, una cota superior ajustada del coste de encontrar una solución utilizando un esquema adecuado es del orden de:

- (a) $O(n^{m/2} \log n)$.
- (b) $O(m \log n)$.
- (c) $O(n^m)$.
- (d) $O(nm^n)$.

3. Indique cuál de las siguientes afirmaciones con respecto a los grafos es **falsa**:

- (a) Un grafo es simple si entre cada par de vértices existe a lo sumo una arista.
- (b) La longitud de un camino es el número de aristas que contiene.
- (c) Un grafo dirigido es conexo si al reemplazar todas sus aristas dirigidas por aristas no dirigidas resulta un grafo conexo no dirigido.
- (d) Se denomina componente conexa de un grafo a un subgrafo conexo maximal.

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

5. Se tiene una tabla hash de tamaño $m=11$ y las funciones $h_1(k) = k \bmod m$ y $h_2(k) = (k \bmod (m - 1)) + 1$. Se pide insertar los valores [22,1,13,11,24,33,18,42,31] mediante doble hashing usando $h(k,i) = (h_1(k) + ih_2(k)) \bmod m$, 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	

6. 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

Problema (4 puntos).

Se distribuyen 32 fichas (*) sobre un tablero como muestra la figura:

```

      # # #
    # * * * #
  # * * * #
# * * * * * #
# * * * * * #
# * * * * * #
  # * * * #
    # * * * #
      # # #
  
```

El juego consiste en mover las fichas hasta que solo quede una, con las siguientes condiciones:

- Las fichas * se mueven saltando una sobre otra, desapareciendo la ficha sobre la que se salta. No se salta en diagonal. Las # indican fin del tablero. No se puede saltar sobre ellas.
- Solo se salta sobre una ficha (y solo una) si tras ella hay un hueco.

Se pide programar un algoritmo que resuelva el problema en el menor número de movimientos.

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

1. Elección del esquema más apropiado, el esquema general y explicación de su aplicación al problema (0,5 puntos).
2. Descripción de las estructuras de datos necesarias (0,5 puntos solo si el punto 1 es correcto).
3. Algoritmo completo a partir del refinamiento del esquema general (2,5 puntos solo si el punto 1 es correcto). Si se trata del esquema voraz, debe realizarse la demostración de optimalidad. Si se trata del esquema de programación dinámica, deben proporcionarse las ecuaciones de recurrencia.
4. Estudio del coste del algoritmo desarrollado (0,5 puntos solo si el punto 1 es correcto).