

1 [40 puntos]

Sea $V=1..n$, para un $n \in \mathbf{nat}$. Una red de comunicaciones se modela con un grafo completo dirigido $G(V, E, c)$ donde, para cada arco $(u, v) \in E$, $c(u, v)$ denota la *confiabilidad* de transmitir de u a v . La confiabilidad es un número real, $0 \leq c(u, v) \leq 1$, que se puede interpretar como la probabilidad de que la transmisión entre los nodos sea exitosa, suponiendo -además- que estas probabilidades son independientes. Este supuesto conlleva el hecho de que la noción de confiabilidad se pueda extrapolar a caminos, de modo que la *confiabilidad de un camino* sea el producto de las confiabilidades de sus arcos.

1a (10/40) Muestre que $S = ([0, 1], \max, *, 0, 1)$ es un semianillo.

1b (20/40) Defina la *confiabilidad* entre $u, v \in V$ como la máxima confiabilidad de los caminos existentes entre u y v . Use 1a para explicar cómo se pueden usar los Algoritmos de Floyd-Warshall y Dijkstra para encontrar un camino de máxima confiabilidad entre todos los puntos $u, v \in V$. Estime las complejidades espacial y temporal de sus respuestas.

1c (10/40) Defina la *confiabilidad de la red* como la mínima confiabilidad entre cualesquier par de nodos u y v . Describa un algoritmo para encontrar la confiabilidad de la red y estime sus complejidades temporal y espacial.

2 [30 puntos]

Una herramienta que analiza programas escritos en un lenguaje OO (v.gr., *Java*) tiene una función que detecta dependencias entre métodos. Considere las siguientes definiciones, dados dos métodos m y m' :

- m *depende de* m' otro si m llama a m' o si m llama a un método que depende de m' .
- m y m' son *mutuamente recursivos* si son diferentes y cada uno depende del otro.

La herramienta cuenta con una función $L(m)$ que recibe como parámetro (el identificador de) un método m y entrega como salida una lista con (los identificadores de los) métodos que m llama, con un costo $O(k)$ donde k es el número de métodos que m llama.

2a (15/30) Considere el problema de decidir si existen métodos mutuamente recursivos en un programa. La entrada se puede imaginar como una lista de n métodos $\langle m_1, m_2, \dots, m_n \rangle$ del programa. Modele el problema como un problema sobre un grafo y explique un algoritmo que lo resuelva. Si usa la función L , indique cómo esto contribuye a su algoritmo.

2b (15/30) Estime las complejidades temporal y espacial de su algoritmo.

3 [40 puntos]

Suponga una secuencia de $n > 0$ números enteros $\langle a_1, a_2, \dots, a_n \rangle$ y un entero z , se quiere averiguar si hay una manera de elegir signos para una suma de la forma

$$\pm a_1 \pm a_2 \pm \dots \pm a_n = z$$

Por ejemplo, con la secuencia $\langle 10, -12, 4, -2 \rangle$ se puede producir 8, ya que $+10 + (-12) + 4 - (-2) = 8$, pero no se puede producir 1.

3a (20/40) Exprese el problema como una búsqueda en grafos.

3b (10/40) Justifique si (i) hay que marcar nodos (ii) hay que verificar que la agenda se vacíe (iii) el algoritmo puede no terminar.

3c (10/40) Estime la complejidad temporal de su algoritmo en términos de n .