Este es el examen final del curso Programación Imperativa Modular (PIMO), 2017-2. El examen tiene 9 preguntas y otorga un total de 70 puntos. El examen es individual y no es permitido el uso de libros ni apuntes.

Nombre, código y grupo: ____________________

Pregunta	1	2	3	4	5	6	7	8	9	**Total**
Puntos	10	10	10	10	10	10	10	10	10	90
Puntaje										

Escoja y resuelva 7 de las 9 preguntas que se formulan a continuación.

1. (10 puntos) Suponga que ha sido contratado para probar la resistencia de unos jarrones. En particular, los experimentos en los cuales Usted participará consisten en arrojar jarrones de cierto tipo desde diferentes alturas: usará una escalera de $n \geq 1$ peldaños y arrojará jarrones hasta determinar cuál es el peldaño más alto desde el cual ese tipo de jarrón no se rompe. Dicho peldaño se denomina como el mejor peldaño.
   a. Para el primer experimento le han entregado $k = 2$ jarrones de tipo J_1. Describa una estrategia que le permita encontrar el mejor peldaño para los jarrones de tipo J_1 que le obligue a hacer a lo sumo $f(n)$ lanzamientos, en donde el crecimiento asintótico de $f(n)$ es menor (estrictamente) que un crecimiento lineal (en función de n).
   b. Para el segundo experimento le han entregado $k > 2$ jarrones de tipo J_2. Describa una estrategia que le permita encontrar el mejor peldaño para los jarrones de tipo J_2 usando a lo sumo k jarrones.

2. (10 puntos) Considere un arreglo A[0..n) de números enteros. Una inversión en A es una pareja de índices $0 \leq i < j < n$ tales que $a_i > a_j$ (es decir, a_i y a_j están desordenados). Por ejemplo, en el arreglo [2, 4, 1, 3, 5] hay tres inversiones: las parejas de índices (0, 2), (1, 2) y (1, 3). Diseñe un algoritmo que determine la cantidad de inversiones en un arreglo de números enteros de longitud n con complejidad temporal O(n log n). Justifique su respuesta.

   Ayuda: Modifique el algoritmo Merge-Sort.

3. (10 puntos) Diseñe una estructura de datos dforest para representar conjuntos disyuntos con ranking. En particular, exhiba pseudo-código para las funciones find(x) y union(x,y) que, respectivamente, calcula el representante de x y realiza la unión de los conjuntos a los cuales x e y pertenecen.

4. Enuncie el Teorema Maestro y utilícelo para calcular la función T(n) en cada uno de los siguientes casos:

   (a) $T(n) = T(n/2) + n^4$
   (b) $T(n) = 2T(n/4) + \sqrt{n}$
   (c) $T(n) = 2T(7n/10) + n$

5. (10 puntos) Sea G = (V; E) un grafo dirigido y suponga que hay una versión de G codificada en una matriz de adyacencia y otra versión de G codificada en listas de adyacencia. Puede suponer que V = {1,…, n}. Exhiba pseudo-código y analice su complejidad temporal, dependiendo de cada una de las codificaciones de G, para cada una de las siguientes operaciones:
   a. Dado un vértice $v \in V$, calcular el grado de llegada de v, es decir, el número arcos que llegan a v.
   b. Determinar si G es simétrico, es decir, calcular si $(u, v) \in E \leftrightarrow (v, u) \in E$ para cualquier par de vértices u y v en V.

6. (10 puntos) Sea $G = (V, E)$ un grafo dirigido. Una bi-coloración de G es una función $c : V \rightarrow \{0, 1\}$ tal que para cualquier par de vértices u y v en V, si $(u, v) \in E$, entonces $c(u) \neq c(v)$. Diseñe un algoritmo que determine si existe una bi-coloración de G.

7. (10 puntos) Sea $G = (V, E)$ un grafo dirigido. Diseñe un algoritmo que determine si en G hay al menos un ciclo.

8. (10 puntos) Sea $G = (V, E)$ un grafo dirigido y $w : E \rightarrow R$ una función de peso. Diseñe un algoritmo que determine si en G hay un ciclo de costo negativo.

9. (10 puntos) Sea $G = (V; E)$ un grafo no dirigido y $w : E \rightarrow R^{\geq 0}$ una función de peso. Diseñe un algoritmo que calcule un árbol de cubrimiento de G de costo mínimo.