



CI-1221 Estructuras de Datos y Análisis de Algoritmos  
I ciclo 2013, grupo 2

IV EXAMEN PARCIAL

*Viernes 12 de julio*  
*1.00 p. m. – 3.30 p. m.*

Nombre: \_\_\_\_\_ Carné: \_\_\_\_\_

Pregunta	Puntos	Páginas	Calificación
1. Análisis amort. y estructuras de datos para conj. disjs.	21		
2. Representación de grafos	10		
3. Recorrido de un grafo en anchura	9		
4. Ord. top. y comps. fuertemente conexos	35		
5. Alg. de Kruskal y estructuras de datos p/conjs. disjs.	20		
6. Algoritmo de Prim	13½		
7. Camino más corto desde una fuente	20		
Total	128½		

El examen consta de 7 preguntas que suman 128½ puntos, pero no se reconocerán más de 130 (30% extra). Cada pregunta indica el tema tratado y su valor. Si la pregunta tiene subítemes, el puntaje de cada uno es indicado dentro de los subítemes. Se recomienda echar un vistazo a los temas de las preguntas y a su puntaje antes de resolver el examen, para así distribuir su tiempo y esfuerzo de la mejor manera. Las preguntas se pueden responder en cualquier orden, pero se debe indicar en el cuadro mostrado en la portada las páginas del cuaderno de examen en la que están las respuestas. Para ello numere las hojas del cuaderno de examen en la esquina superior externa de cada página. El examen se puede realizar con lápiz o lapicero. No se permite el uso de dispositivos electrónicos (calculadoras, teléfonos, audífonos, etc.).

1. Análisis amort. y estructuras de datos para conj. disjs. [21 pts.]

Suponga que se desea unir  $n = 2^k$  conjuntos de cardinalidad uno, creados mediante la operación MAKE-SET e implementados mediante árboles. Para ello Robert (Prim) ha propuesto que se unan primero dos conjuntos y que luego se le una a estos un tercero, luego un cuarto, etc. Joseph

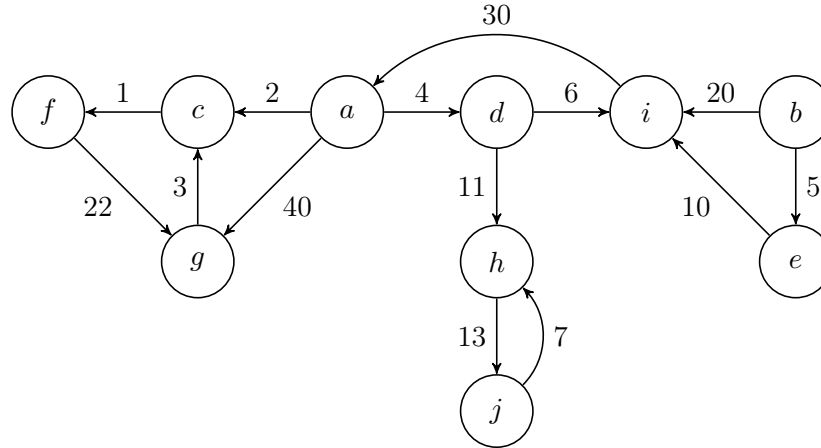


Figura 1: Grafo para las preguntas 3, 4, 5, 6 y 7.

(Kruskal), en cambio, ha propuesto que se formen parejas y que luego estas se unan en cuartetos, luego en octetos, etc. Asumiendo que los conjuntos son implementados mediante árboles y que los argumentos dados al método UNION son escogidos cuidadosamente para minimizar el tiempo de ejecución de las uniones:

- Calcule el tiempo amortizado que toma unir todos los conjuntos siguiendo la sugerencia de Robert. [8 pts.]
- Calcule el tiempo amortizado que toma unir todos los conjuntos siguiendo la sugerencia de Joseph. [12 pts.]
- Indique cuál método es más eficiente. [1 pto.].

2. *Representación de grafos* [10 pts.]

Suponga que se desea determinar si existe un camino  $\langle v_1, v_2, \dots, v_k \rangle$  en un grafo con  $V$  vértices y  $A$  aristas.

- ¿Qué representación de grafo —listas o matriz de adyacencia— permite determinar más rápidamente si existe el camino? Justifique su respuesta. [5 pts.]
- Basado en su respuesta al punto a), indique cuánto tiempo toma en el peor caso determinar la existencia del camino. [5 pts.]

3. *Recorrido de un grafo en anchura* [9 pts.]

Realice un recorrido a lo ancho del grafo mostrado en la figura 1. Parta del vértice  $a$  e ignore los pesos de las aristas. Asuma que las listas de adyacencia están ordenadas alfabéticamente. Muestre al finalizar cada iteración principal del algoritmo lo siguiente:

- La distancia del vértice  $a$  a cada uno de los vértices. [3 pts.]
- Los predecesores que forman el camino correspondiente. [3 pts.]
- El estado de la cola. [3 pts.]

Si al finalizar una iteración no muestra la distancia ni el predecesor de un nodo, se asumirán los mismos que en la iteración anterior ( $\infty$  y NIL, respectivamente, si no se han especificado aún).

4. *Ord. top. y comps. fuertemente conexos* [35 pts.]

- a) Haga un recorrido en profundidad del grafo de la figura 1 y muestre los tiempos de descubrimiento y finalización [5 pts.] y vértices predecesores [5 pts.] correspondientes. Asuma que las listas de adyacencia están ordenadas alfabéticamente y que en el ciclo principal los vértices son tomados en orden alfabético. Ignore los pesos de las aristas.
- b) Etiquete las aristas que no pertenecen a un árbol como aristas hacia adelante (F), aristas hacia atrás (B) o aristas cruzadas (X). [4 pts.]
- c) Según el etiquetado realizado en el punto anterior, indique qué aristas deben ser eliminadas del grafo para poder realizar un ordenamiento topológico [2 pts.]. Efectúe tal ordenamiento [5 pts.].
- d) Dibuje la traspuesta del grafo e ignore los pesos de las aristas. [2 pts.]
- e) Haga un recorrido en profundidad de la traspuesta del grafo tomando los vértices en cada visita de acuerdo a los tiempos de finalización obtenidos en el punto (a). Muestre los respectivos tiempos de descubrimiento y finalización [5 pts.] y predecesores [5 pts.]. Utilice esta información para encontrar los componentes fuertemente conexos del grafo [2 pts.].

5. *Alg. de Kruskal y estructuras de datos p/conjs. disjs.* [20 pts.]

Utilice el algoritmo de Kruskal para encontrar un árbol recubridor mínimo para la versión no dirigida del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). Para ello, tome en cuenta que los pesos de las aristas, ordenados de menor a mayor son los siguientes: 1, 2, 3, 4, 5, 6, 7, 10, 11, 13, 20, 22, 30 y 40. Muestre para cada iteración lo siguiente:

- La arista considerada, y si fue tomada o no. [5 pts.]
- El estado de la correspondiente estructura de datos para conjuntos disjuntos, implementada mediante árboles. Indique el rango de los nodos por medio de un subíndice. [15 pts.]

Para esta pregunta asuma que los argumentos de la operación UNION le son dados en orden alfabético (por ejemplo, como UNION( $a, b$ ) y no UNION( $b, a$ )). Si al finalizar una iteración no muestra el rango ni el predecesor de un nodo, se asumirá que son los mismos que en la iteración anterior (0 y NIL, respectivamente, si no se han especificado aún).

6. *Algoritmo de Prim* [ $13\frac{1}{2}$  pts.]

Utilice el algoritmo de Prim para encontrar un árbol recubridor mínimo de la versión no dirigida del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). Especifique en cada paso las aristas consideradas y la arista seleccionada. Empiece en el vértice  $a$ .

7. *Camino más corto desde una fuente* [20 pts.]

Utilice el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice  $b$  y los otros vértices en grafo de la figura 1. Muestre para cada iteración lo siguiente:

- Las distancias más cortas conocidas al vértice  $b$ . [10 pts.]
- Los predecesores que forman tal camino. [10 pts.]

Si al finalizar una iteración no indica la distancia o el predecesor de un vértice, se asumirá que son los mismos que en la iteración anterior (o  $\infty$  y NIL, respectivamente, si no se han especificado aún).