



CI-1221 Estructuras de Datos y Análisis de Algoritmos
II ciclo 2012, grupos 2 y 3

IV EXAMEN PARCIAL

Sábado 1 de diciembre
9.00 a. m. – 11.30 a. m.

Apellido(s): _____ Nombre: _____ Carné: _____

Pregunta	Puntos	Páginas	Calificación
1. <i>Análisis amortizado</i>	15		
2. <i>Representación de grafos</i>	11		
3. <i>Recorrido de un grafo a lo ancho</i>	9		
4. <i>Ord. top. y comps. fuertemente conexos</i>	35		
5. <i>Alg. de Kruskal y estructuras de datos p/conjs. disjs</i>	20		
6. <i>Algoritmo de Prim</i>	13½		
7. <i>Camino más corto desde una fuente</i>	20		
Total	123½		

El examen consta de 7 preguntas que suman 123½ 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 de ellos 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 la tabla mostrada en la página 1 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 amortizado.* [15 pts.]

Calcule el costo amortizado de insertar los elementos $0, 1, 2, \dots, n-1$, en ese orden, en una tabla *hash* vacía de tamaño $m = n$, con función hash auxiliar $h'(k) = k \bmod m$. Utilice, para ello cada una de las siguientes técnicas de resolución de colisiones.

a) Encadenamiento: $h(k) = h'(k)$. [4 pts.]

b) Direccionamiento abierto con sondeo lineal:

$$h(k, i) = (h'(k) + i) \bmod m \quad (i = 0, 1, \dots, m-1). \text{ [5 pts.]}$$

c) Direccionamiento abierto con sondeo cuadrático:

$$h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m \quad (i = 0, 1, \dots, m-1),$$

donde c_1 y c_2 han sido cuidadosamente seleccionados para hacer uso de todas las posiciones de la tabla. [6 pts.]

2. *Representación de grafos.* [11 pts.]

Suponga que se desea encontrar en un grafo con V vértices y A aristas el vértice con mayor cantidad de adyacentes.

- ¿Cuánto tiempo toma encontrar tal vértice en un grafo representado mediante listas de adyacencia? Explique. [5 pts.]
- ¿Cuánto tiempo toma encontrarlo en un grafo representado mediante una matriz de adyacencia? Explique. [5 pts.]
- De acuerdo a su respuestas anteriores, ¿que representación de grafo permite encontrar el vértice con mayor cantidad de adyacentes más rápidamente? [1 pto.]

3. *Recorrido de un grafo a lo ancho.* [9 pts.]

Realice un recorrido a lo ancho del grafo mostrado en la figura 1, partiendo del vértice a e ignorando 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 asumirá que son los mismos de la iteración anterior (o ∞ y NIL, respectivamente, si no se han especificado).

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

- Haga un recorrido en profundidad del grafo de la figura 1 y muestre los tiempos de descubrimiento y finalización [5 pts.] y predecesores [5 pts.] correspondientes. Asuma que las listas de adyacencia están ordenadas alfabéticamente y que en el ciclo principal los vértices son considerados en orden alfabético. Ignore los pesos de las aristas.
- Etiquete las aristas que no pertenecen a un árbol como aristas hacia adelante (F), aristas hacia atrás (B) o aristas cruzadas (X). [4 pts.]
- Según el etiquetado realizado en el punto anterior, indique qué aristas deben ser eliminadas del grafo para poder realizar un ordenamiento topológico [2 pts.] y efectúe tal ordenamiento [5 pts.].
- Dibuje la traspuesta del grafo, ignorando los pesos de las aristas. [2 pts.]
- 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: 5, 10, 13, 14, 15, 20, 22, 25, 26, 30, 33, 40 y 100.

Muestre en cada iteración lo siguiente:

- La arista considerada, y si fue tomada o no. [5 pts.]

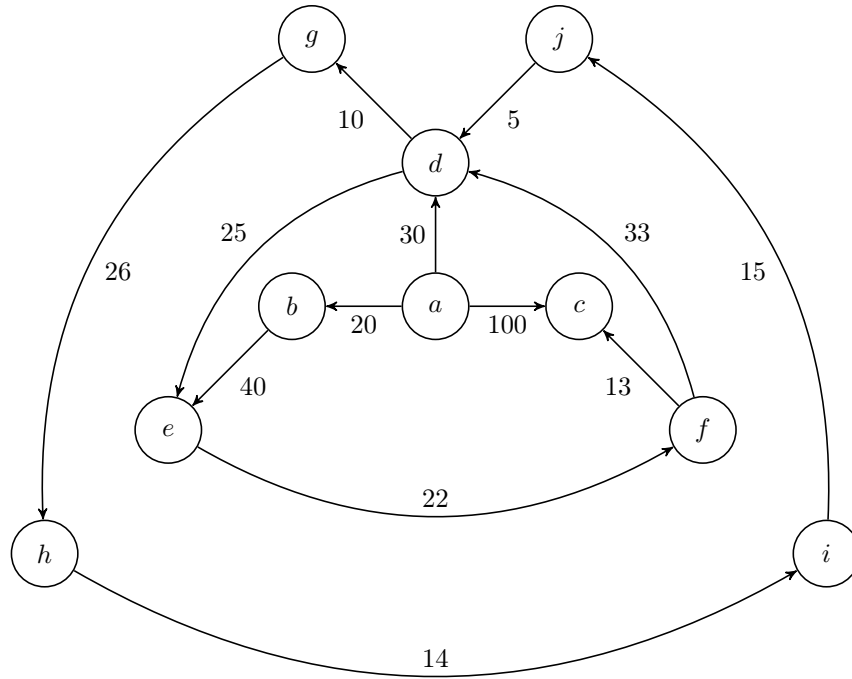


Figura 1: Grafo para las preguntas 3, 4, 5, 6 y 7.

- El estado de la correspondiente estructura de datos para conjuntos disjuntos implementada mediante árboles, incluyendo el rango de los nodos (como subíndices). [15 pts.]

Asuma que los argumentos le son dados a la operación UNION 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 de la iteración anterior (o 0 y NIL, respectivamente, si no se han especificado aún).

6. *Algoritmo de Prim.* [13½ 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 a y los otros vértices del grafo de la figura 1.

Muestre en cada iteración lo siguiente:

- Las distancias más cortas conocidas al vértice a . [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 ∞ y NIL, respectivamente, si no se han especificado aún).