



CI-1221 Estructuras de Datos y Análisis de Algoritmos
I ciclo 2011, Grupo 03

IV EXAMEN PARCIAL

Martes 5 de julio

El examen consta de 9 preguntas que suman $103\frac{1}{2}$ puntos, pero no se reconocerán más de 130 (30 % extra). Cada pregunta empieza con un indicador de su puntaje y del tema tratado. Si la pregunta tiene subítemes, el puntaje de cada uno de ellos es indicado dentro de los subítemes. Las preguntas se pueden responder en cualquier orden pero el número de pregunta debe ser claramente indicado. El examen se puede realizar con lápiz o lapicero. **No se permite el uso de calculadora.**

1. [20 pts.] *Análisis amortizado*

Suponga que una lista enlazada contiene las llaves $1, 2, \dots, n$, en ese orden, y que se hacen k búsquedas (independientes) en ella, correspondiendo cada búsqueda a una de las distintas potencias de 2 menores que n (es decir: $1, 2, 4, 8, \dots, 2^{\lfloor \lg(n-1) \rfloor}$). Determine una cota asintótica, lo más ajustada posible, del tiempo de ejecución total [15 pts.] y amortizado [5 pts.] que toma hacer las k búsquedas. (Si le facilita las cosas, puede asumir que n es una potencia de 2).

2. [18 pts.] *Estructuras de datos para conjuntos disjuntos*

Simule la ejecución del conjunto de operaciones mostrado abajo. Muestre el estado de cada árbol modificado con cada llamado a MAKE-SET, UNION, LINK ó FIND-SET (incluyendo los llamados a FIND-SET que se hacen dentro de UNION). Indique además para cada nodo: su padre (por medio de una flecha a él) y su rango (escribiéndolo al lado del nodo [si lo prefiere, solo cuando cambia de valor]). **Recuerde que, por convención, cuando los rangos son iguales, el árbol derecho “adopta” al izquierdo, y no al revés.**

1	for x = a,b,...,h	
2	MAKE-SET(x)	[2 pts.]
3	UNION(a,b)	[1 pto.]
4	UNION(c,d)	[1 pts.]
5	UNION(e,f)	[1 pts.]
6	UNION(g,h)	[1 pts.]
7	UNION(a,c)	[2 pto.]
8	UNION(a,e)	[4 pto.]
9	UNION(e,g)	[4 pts.]

3. [5 pts.] *Representación de grafos*

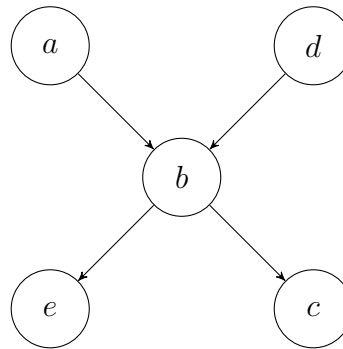
Determine una cota asintótica lo más ajustada posible para el espacio requerido por un grafo $G = (V, A)$ representado como:

- a) Una lista de adyacencia. [2.5 pts.]
- b) Una matriz de adyacencia. [2.5 pts.]

4. [5 pts.] *Representación de grafos*

Represente el grafo siguiente como:

- a) Una lista de adyacencia. [2½ pts.]
- b) Una matriz de adyacencia. [2½ pts.]



5. [7½ pts.] *Recorrido de un grafo a lo ancho*

Realice un recorrido a lo ancho del grafo de la pregunta 4, partiendo del vértice a , y asumiendo que las listas de adyacencia se encuentran ordenadas alfabéticamente. Debe quedar especificada la distancia del vértice a a cada uno de los vértices [2½ pts.], los predecesores que forman el camino correspondiente [2½ pts.] y el orden en que los vértices fueron insertados en la cola [2½ pts.].

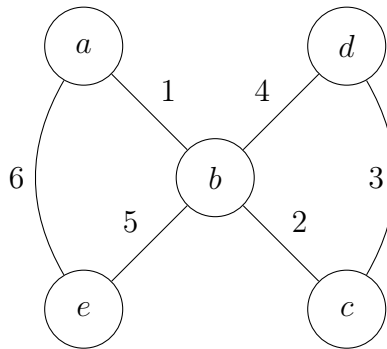
6. [14 pts.] *Recorrido de un grafo en profundidad*

- a) Realice un recorrido en profundidad del grafo mostrado en la pregunta 4. Indique los tiempos de descubrimiento y finalización de cada uno de los vértices [5 pts.], el predecesor de cada uno de los vértices [2½ pts.] y la clasificación de las aristas que no pertenezcan a los árboles (hacia atrás [B], hacia adelante [F], cruzadas [C]) [1½ pts.]. Asuma que el ciclo principal considera los vértices en orden lexicográfico y que las listas de adyacencia son dadas en orden lexicográfico también.
- b) Utilice los resultados de la parte anterior para producir un ordenamiento topológico del grafo. [5 pts.]

7. [12 pts.] *Componentes fuertemente conexos*

- a) Dibuje la transpuesta del grafo mostrado en la pregunta 4. [2 pts.]

- b) Haga un recorrido en profundidad de esta transpuesta, tomando los vértices en cada visita de acuerdo a la ordenación topológica producida en la pregunta 6. Especifique los tiempos de descubrimiento y finalización [5 pts.] y los predecesores de cada uno de los vértices producidos por el recorrido [$2\frac{1}{2}$ pts.] y utilice esta información para determinar los componentes fuertemente conexos del grafo [$2\frac{1}{2}$ pts.].
8. [12 pts.] *Árboles recubridores mínimos*
 Utilice el algoritmo de Kruskal [6 pts.] y el algoritmo de Prim [6 pts.] para encontrar los árboles recubridores mínimos en el siguiente grafo. **Especifique para cada algoritmo las aristas que fueron *consideradas* en cada iteración (incluyendo las aristas consideradas pero rechazadas).** Para el algoritmo de Prim empiece en la arista a .



9. [10 pts.] *Camino más corto desde una fuente*
 Utilice el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice a y los otros vértices en el siguiente grafo. **Muestre las distancias [5 pts.] y los predecesores [5 pts.] de cada vértice después de cada iteración principal del algoritmo.**

