

CI-1221 Estructuras de Datos y Análisis de Algoritmos
II ciclo 2011, Grupo 02

IV EXAMEN PARCIAL

Lunes 28 de noviembre

El examen consta de 8 preguntas que suman $134\frac{1}{2}$ puntos, pero no se reconocerán más de 125 (25 % 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 abajo los números de página del cuaderno de examen en las que está cada respuesta. Para ello las hojas del cuaderno de examen deben estar numeradas 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.).*

Pregunta	Puntos	Páginas	Calificación
1. <i>Análisis amortizado</i>	20		
2. <i>Estructuras de datos para conjuntos disjuntos</i>	$13\frac{1}{2}$		
3. <i>Representación de grafos</i>	6		
4. <i>Recorrido de un grafo a lo ancho y en profundidad</i>	25		
5. <i>Ordenamiento topológico</i>	21		
6. <i>Componentes fuertemente conexos</i>	23		
7. <i>Árboles recubridores mínimos</i>	14		
8. <i>Camino más corto desde una fuente</i>	12		
Total	$134\frac{1}{2}$		

1. *Análisis amortizado.* [20 pts.]

Suponga que se tiene una estructura de datos que cada vez que se llena incrementa su tamaño por un factor $\sqrt{2}$, y que tal incremento del tamaño toma tiempo proporcional al tamaño (el viejo o el nuevo, da lo mismo). Determine el comportamiento asintótico del tiempo total [15 pts.] y amortizado [5 pts.] que toma hacer n inserciones en tal estructura.

2. Estructuras de datos para conjuntos disjuntos. [13½ pts.]

Simule la ejecución del conjunto de operaciones mostrado abajo. Muestre el estado de cada árbol modificado con cada llamado a MAKE-SET, UNION, LINK o FIND-SET (incluyendo los llamados a FIND-SET hechos 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 para  $x = a \dots i$ 
2   MAKE-SET( $x$ )
3 UNION( $b, d$ )
4 UNION( $e, g$ )
5 UNION( $a, c$ )
6 UNION( $h, i$ )
7 UNION( $a, b$ )
8 UNION( $e, f$ )

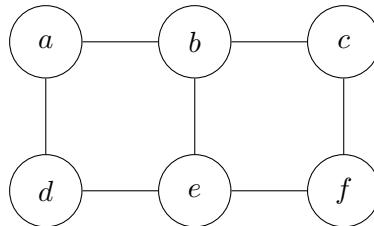
```

[½ pts.]

3. Representación de grafos. [6 pts.]

Represente el siguiente grafo utilizando:

- Listas de adyacencia, donde cada lista se ordena alfabéticamente. [3 pts.]
- Una matriz de adyacencia. [3 pts.]



4. Recorrido de un grafo a lo ancho y en profundidad. [25 pts.]

A continuación se muestra una versión modificada del algoritmo de búsqueda a lo ancho (BFS) en el que no se guardan ni las distancias ni los predecesores del árbol generado, sino que simplemente se imprimen los vértices ordenados de acuerdo a su distancia a la fuente (primero la fuente, luego los vértices a una distancia de 1, luego los vértices a una distancia de 2, etc.):

BFS(G, f)

```

1 para cada  $v \in G.V - \{f\}$ 
2    $v.color = \text{BLANCO}$ 
3  $f.color = \text{GRIS}$ 
4  $C = \emptyset$ 
5 ENCOLAR( $C, f$ )
6 mientras  $C \neq \emptyset$ 
7    $v = \text{DESENCOLAR}(C)$ 
8   imprima  $v$ 
9    $v.color = \text{NEGRO}$ 
10  para cada  $w \in Adj[v]$ 
11    si  $w.color = \text{BLANCO}$ 
12       $w.color = \text{GRIS}$ 
13      ENCOLAR( $C, w$ )

```

- a) Modifique el código para que el recorrido se haga en profundidad en vez de a lo ancho. El procedimiento debe ser iterativo (*no recursivo*). [8 pts.]

Nota. *No es necesario visitar vértices no alcanzables desde la fuente.*

Sugerencias:

- I. Considere utilizar otra estructura de datos en vez de una cola.
 - II. Si es conveniente, utilice más o menos colores.
- b) Simule la ejecución de su algoritmo sobre el grafo de la pregunta 3, mostrando al finalizar cada una de las iteraciones del ciclo principal (el ciclo **mientras**) lo siguiente:

- I. La salida.
- II. El estado de la estructura de datos auxiliar (si utiliza alguna).
- III. El coloreo del grafo.

Para esto asuma que la lista de nodos adyacentes de cada vértice se encuentra ordenada alfabéticamente. [12 pts.]

- c) Determine la complejidad del algoritmo y argumente su respuesta. [5 pts.]

5. *Ordenamiento topológico.* [21 pts.]

A continuación se listan las tareas que el prof. Camacho debe realizar para preparar el desayuno favorito de su hija, *pancakes*, junto con su bebida favorita (de él, no de ella), jugo de naranja. (© 2011 A. Camacho).

- A. Partir un huevo y ponerlo en un plato hondo (mismo plato de la tarea B).
- B. Poner en un plato hondo (mismo plato de la tarea A) harina, yogur, vainilla y nuez moscada.
- C. Revolver contenidos del plato hondo.
- D. Poner sartén a calentar.
- E. Poner aceite en la sartén y esperar a que caliente.
- F. Verter la mezcla en la sartén formando pancakes de unos 5 cm de radio, rociar canela y esperar a que se doren.
- G. Voltear los pancakes y esperar a que se doren.
- H. Servir los pancakes y ponerlos en la mesa.
- I. Tomar cuatro naranjas.
- J. Lavar las naranjas.
- K. Partir las naranjas.
- L. Exprimir las naranjas.
- M. Servir el jugo en vasos y ponerlos en la mesa.
- N. Sentarse a la mesa.
- Ñ. Agregar sirope a los pancakes y degustar.

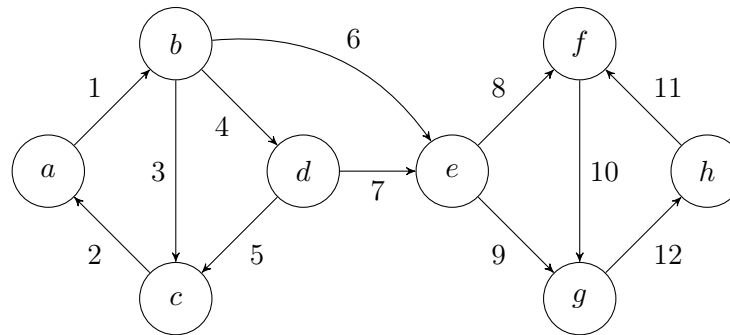
A partir de la lista anterior y su experiencia culinaria (o en su ausencia, sentido común):

- a) Construya un grafo que muestre todas las dependencias *reales* existentes entre las tareas (no cree dependencias que no se puedan inferir del texto o que no correspondan al sentido común). [7 pts.]

- b) Realice un recorrido en profundidad del grafo construido en el punto anterior, indicando los tiempos de descubrimiento y finalización de cada uno de los vértices [7 pts.] y el predecesor de cada uno de los vértices [3½ pts.]. Asuma que el ciclo principal considera los vértices en el orden alfabético y que las listas de adyacencia siguen ese orden también.
- c) Utilice los resultados de la parte anterior para producir un ordenamiento topológico del grafo que le muestre al profesor el orden en que debe realizar las tareas para preparar el desayuno. [3½ pts.]

6. *Componentes fuertemente conexos.* [23 pts.]

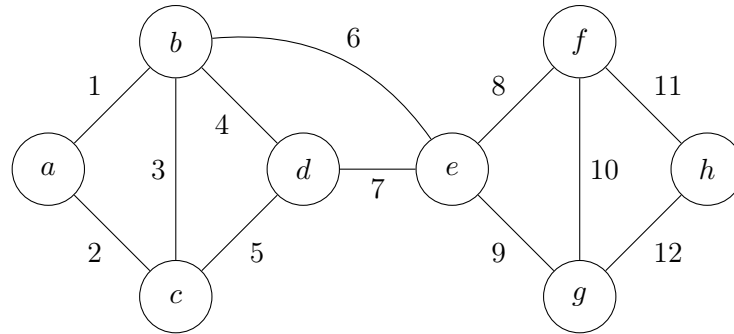
Considere el siguiente grafo:



- a) Haga un recorrido en profundidad del grafo y muestre los tiempos de descubrimiento y finalización [4 pts.] y predecesores [2 pts.] correspondientes. Asuma que las listas de adyacencia están ordenadas alfabéticamente y que en el ciclo principal los vértices son considerados en ese orden también.
- b) Etiquete las aristas que no pertenecen a los árboles producidos por el recorrido del punto anterior como aristas hacia adelante (F), aristas hacia atrás (B) o aristas cruzadas (C). [5 pts.]
- c) Dibuje la traspuesta del grafo. [3 pts.]
- d) 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 [4 pts.] y predecesores [2 pts.] y utilice esta información para mostrar los componentes fuertemente conexos del grafo [3 pts.].

7. *Árboles recubridores mínimos.* [14 pts.]

Utilice los algoritmos de Kruskal [7 pts.] y Prim [7 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 *h*.



8. *Camino más corto desde una fuente.* [12 pts.]

Utilice el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice a y los otros vértices en el grafo de la pregunta 6. **Muestre las distancias [6 pts.] y los predecesores [6 pts.] de cada vértice después de cada iteración principal del algoritmo.**