Algoritmos y Estructuras de Datos
Tarea 8

Profesor: Sergio Rajsbaum, Ayudante: David Flores

fecha de hoy: 24 de noviembre 2005, **fecha de entrega:** 6 de diciembre 2005
– No se aceptan tareas después de esta fecha
— *explica en detalle y con claridad* <u>*todas*</u> *tus respuestas* —
Se permite trabajar en equipos de hasta DOS personas.
Pero cada uno debe entregar la tarea resuelta por separado, e indicar el nombre de su compañero de equipo.

Tema: Heaps de Fibonacci, AGM, DFS

1. Escribe un resumen de no más de una cuartilla del capítulo de Lamport (página 120) y otro del capítulo de Cook y Levin (página 139) del libro *Out of their Minds* de Shasha y Lazere.

2. En un Heap de Fibonacci, vimos en clase que el i-ésimo hijo de cualquier vértice tiene rango al menos $i-2$. Sea F_n el árbol más pequeño posible the rango n que satisface esta propiedad. El *rango* de un árbol es el número de hijos de su raíz. Sea $|F_n|$ el número de vértices de F_n. Vimos que: $|F_0| = 1$, $|F_1| = 2$, $|F_2| = 3$, $|F_3| = 5$, $|F_4| = 8$, $|F_5| = 13$. Y los árboles correspondientes están en la siguiente figura.

   (a) Describe como es F_6, y cuanto vale $|F_6|$.

   (b) Explica como se puede llegar a tener tu árbol F_6 del inciso anterior.

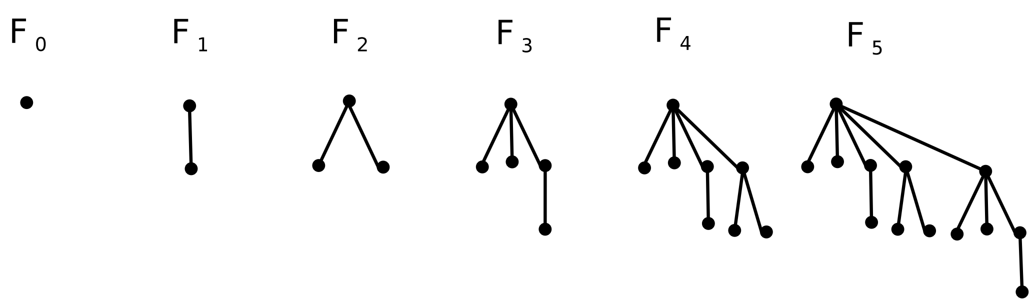

Figure 1: Los primeros seis F_i's

3. Sea $G = (V, E)$ una digráfica. Demuestra que las dos siguientes proposiciones son equivalentes (sin usar el lema correspondiente visto en clase)

(a) La gráfica no dirigida de G es conexa y G tiene un vértice r para el cual ingrado$(r) = 0$, mientras que para cualquier otro vértice v distinto de r ingrado$(v) = 1$

(b) G tiene una raíz r en V desde la cual se puede llegar a los demás vértices por medio de un único camino dirigido.

4. Sea $G = (V, E)$ una gráfica con pesos en las aristas ℓ_e para cada arista e. Estamos interesados en construir una subgráfica $H = (V, F)$, tal que la longitud del camino más corto entre cualquier pareja u, v en H no sea mucho más largo que el camino más corto entre u, v en G. La longitud de un camino se refiere a la suma de los pesos de sus aristas. A cambio, toleramos que H tenga un poco más de aristas que un árbol generador de G. Para simplificar las cosas, supongamos que no hay dos aristas con el mismo peso. Considera la siguiente variante del algoritmo de Kruskal.

   - Primero ordenamos todas las aristas de acuerdo a su peso, de menor a mayor.
   - Luego construimos una subgráfica $H = (V, F)$ considerando cada arista en orden:
   - Al considerar una arista $e = (u, v)$, añadimos e a H si hasta este momento no hay un camino entre u y v en H. Pero si hay un camino tal, entonces sea $d_H(u, v)$ la longitud del menor entre estos caminos de u a v. Añadimos e a H si $3\ell_e < d_H(u, v)$.

   Es decir, añadimos la arista e solamente si reduce la distancia entre u y v suficientemente.

   Sea $H = (V, F)$ la subgáfica de G que regresa el algoritmo.

   (a) Demuestra que para cada pareja de vértices u, v, la longitud del camino más corto en H, $d_H(u, v)$ es a lo más 3 veces la longitud del camino más corto en G, $d_G(u, v)$.

   (b) La gráfica H obtenida no puede ser mucho más densa que un árbol generador. Sea $f(n)$ el máximo número de aristas que H puede llegar a tener, sobre cualquier gráfica G de n vértices. Demuestra que
   $$\lim_{n\to\infty} \frac{f(n)}{n^2} = 0$$
   Tip: Prueba que H no puede tener ciclos de longitud 3 o 4, y luego que el grado mínimo en esta gráfica no puede ser mayor a $\sqrt{n}$. Finalmente prueba que el número de aristas de H es a lo más $n^{3/2}$.

5. Sea $G = (V, E)$ una digráfica. Decimos que $u \equiv v$ si u y v pertenecen a algún ciclo dirigido de G.

   (a) Demuestra que esta relación $\equiv$ es de equivalencia

   (b) Las clases de equivalencia se llaman *componentes fuertemente conexas* de G. Describe que propiedades debe tener la gráfica de super-estructura de G; es decir, la gráfica que contiene un vértice por cada componente fuertemente conexa, y un arco de una componente u a otra v si existe un arco en G de la componente que u representa a la que v representa.

   (c) Describe un algoritmo de tiempo $O(|E|)$ para detectar las componentes fuertemente conexas de G. Tip: se puede usar una variante de DFS para gráficas dirigidas.

Para cualquier duda contactar a David o a Sergio a colegadavid@gmail.com, rajsbaum@matem.unam.mx con suficiente anticipación.