Algoritmos y Estructuras de Datos
Tarea 7

Profesor: Sergio Rajsbaum
Ayudante: Francisco Javier Escalona González

fecha de hoy: 25 de noviembre 2004
fecha de entrega: 13 de diciembre 2004
— *explica en detalle y con claridad todas tus respuestas* —
— *Tus algoritmos deberán ser lo más eficiente posibles* —
— *explica el funcionamiento de tus algoritmos informalmente, luego escribe el código, y luego demuestra correctez y complejidad.* —
Se permite trabajar en equipos de dos personas.
Pero cada uno debe entregar la tarea resuelta por separado, e indicar el nombre de su compañero de equipo.

Tema: AGM, distancia, flujo

1. Escribe el código de la rutina BUILD_HEAP(A) vista en clase y demuestra *en detalle* (incluyendo la prueba de cualquier ecuación que utilizes) que su complejidad es $O(n)$.

2. Demuestra que el algoritmo de Kruskal es correcto.

3. Considera el siguiente algoritmo recursivo, $agm(G, w)$ que toma como entrada una gráfica conexa $G = (V, E)$ con pesos en las aristas w. Se utiliza la operación $G' = G - e$, que resulta en la gráfica obtenida al contraer una arista $e = (u, v)$; es decir, borrar e y reemplazar u, v por un solo vértice u, y w la misma función de pesos restringida a G'. Demuestra que el algoritmo correctemane produce un árbol generador de peso mínimo y analiza su complejidad. ¿ Qué sucede si G no es conexa?

```
Function agm(G = (V, E), w)
        if |V| = 1 then return() fi
        if E = {e} then return(E) fi
        sea e = (u, v) la arista de menor peso de G
        sea G' = G − e
        return ({e, agm(G', w)}).
```

Figure 1: Algoritm recursivo agm

4. Diseña un algoritmo recursivo similar al del inciso anterior, pero que sea equivalente al algoritmo de Prim.

5. Presenta un ejemplo de una digráfica $G = (V, E)$ con pesos w y origen $s \in V$ tal que G satisface la siguiente propiedad: para todo arco $u \to v \in E$ existe un árbol de caminos más

cortos con raíz en s que contiene a $u \to v$ y otro árbol de caminos más cortos con raíz en s que no contiene a $u \to v$.

6. Demuestra que el requerimiento de que los pesos en las aristas sean no-negativos es indispensable para la correctez del algoritmo de Dijkstra.

7. Describe un algoritmo para encontrar el número de caminos más cortos de un vértice s a un vértice t en una digráfica G sobre la cual se ha ejecutado el algoritmo de Dijkstra. Asume que los pesos en las aristas son positivos. ¿Por qué es necesaria esta suposición?

8. Como fue visto en clase, el algoritmo de Floyd requiere de espacio $\Theta(|V|^3)$. Modifica el algoritmo para que requiera nada mas espacio $\Theta(|V|^2)$.

9. La *cerradura transitiva* de una digráfica $G = (V, E)$ es una digráfica $G' = (V, E')$ tal que el arco $u \to v$ esta en E' si y solo si hay un camino dirigido (no vacío) de u a v en G. Para los algoritmos BFS, Dijkstra, Ford, Johnson y Floyd, explica como pueden ser usados para construir G' a partir de G y compara las complejidades de los algoritmos resultantes. Explica para que gráficas (según cantidad de aristas) conviene usar que algoritmos.

10. Sea $G = (V, E)$ una digráfica con pesos w (positivos o negativos). Describe un algoritmo de tiempo $O(|V||E|)$ para hallar el valor de $\delta^*(v)$ para todo $v \in V$, donde $\delta^*(v) = \min_{u \in V}\{\delta(v, u)\}$. (pista: añade un nuevo vértice s a G con arcos de s a todos los demás.)

11. Presenta un ejemplo de una red de flujo $G = (V, E)$ con fuente s y destino t tal que el algoritmo de Edmonds-Karp ejeccute al menos $2|E|$ iteraciones de caminos aumentantes al correrlo sobre G

No se aceptan tareas despues de la fecha límite.

Para cualquier duda contactar a Paco o a Sergio con suficiente anticipación:
pax@ciencias.unam.mx, rajsbaum@matem.unam.mx