CI7621 – Tarea #3

Prof. Blai Bonet

Diciembre 18/2013 — Enero 8/2014

Resuelva 5 problemas de la siguiente lista. Su selección debe contener al menos dos problemas en $\{3, 5, 6, 7\}$.

1.- Sea $G = (V, E)$ un grafo no dirigido y conectado. De un algoritmo que corra en tiempo $O(V + E)$ para computar un camino en G que recorre cada arista en E exactamente dos veces: una vez en cada dirección. Describa como utilizar dicho algoritmo para encontrar la salida de un laberinto si se tiene disponible una gran cantidad de "migajas de pan".

2.- Un grafo dirigido $G = (V, E)$ es *simplemente conectado* si para cada par de vértices $u, v \in V$, G contiene a lo sumo un camino simple $u \rightsquigarrow v$. De un algoritmo eficiente para determinar cuando un grafo dirigido es simplemente conectado.

3.- Un grafo $G = (V, E)$ es *parcialmente dirigido* si las aristas se particionan en $E = E_1 \cup E_2$ donde E_1 es un conjunto de aristas dirigidas y E_2 es un conjunto de aristas no dirigidas. Una orientación de G es un grafo dirigido $G' = (V, E')$ tal que $E_1 \subseteq E'$ y las aristas en $E' \setminus E_1$ son precisamente las aristas en E_2 que han sido dirigidas de alguna forma. De un algoritmo de tiempo lineal que determine si un grado parcialmente dirigido G tiene un orientación G' que es acíclica.

4.- De un algoritmo para determinar cuando un grafo no dirigido $G = (V, E)$ contiene un ciclo. Su algoritmo debe correr en tiempo $O(V)$, independiente del número $|E|$ de aristas.

5.- Un *tour euleriano* en un grafo $G = (V, E)$ (dirigido y fuertemente conectado) es un ciclo que recorre cada arista de G exactamente una vez, aunque puede visitar un mismo vértice múltiples veces.

a) Muestre que G tiene un tour euleriano si y sólo si para cada vértice $v \in V$, in-degree(v) = out-degree(v).

b) Describa un algoritmo que corre en tiempo $O(E)$ para conseguir un tour euleriano en G si existe. (Ayuda: Una ciclos formados por conjuntos disjuntos de aristas.)

6.- Sea $G = (V, E)$ un grafo dirigido con pesos o costos $w : E \to \mathbb{R}$, y suponga que G tiene un ciclo de costo negativo. De un algoritmo eficiente para listar los vértices en un tal ciclo.

7.- De un algoritmo eficiente para contar el número total de caminos en un grafo dirigido acíclico.

8.- De un ejemplo de una red de flujo $G = (V, E)$ con capacidades $c : E \to \mathbb{R}$ para el cual el algoritmo Ford-Fulkerson puede no terminar. Debe especificar el grafo, las capacidades y una secuencia infinita de caminos de aumento para la cual Ford-Fulkerson no termina.