Ma47a Optimización Combinatorial	**8 de Abril, 2006**

Control No. 1

Prof. Cátedra: M. Kiwi — *Prof. Auxiliar: G. Sanchez*

TIEMPO: 4.5 HRS.

PROBLEMA 1:

(i).- En cátedra se probaron las siguientes afirmaciones:

(i.1).- (1.5 pts) Si $G = (V, E)$ es conexo, entonces $|E| \geq |V| - 1$.
(i.2).- (1.5 pts) Si $G = (V, E)$ no posee circuitos, entonces $|E| \leq |V| - 1$.

Rehaga las demostraciones de ambas implicancias.

(ii).- (3.0 pts) Una digrafo $D = (V, A)$ se dice orientación Euleriana de un grafo (no-dirigido) G si para todo $v \in V$,

$$|\{uv \in A \,:\, u \in V\}| = |\{vu \in A \,:\, u \in V\}| \,.$$

Muestre que dado un grafo (no-dirigido) $G = (V, E)$ en que todo nodo tiene grado par, se puede determinar una orientación Euleriana de G en tiempo $O(|V|+|E|)$ (indique sobre que representación del grafo opera su algoritmo).

PROBLEMA 2: Sea G un grafo (no–dirigido) conexo con función de costos en los arcos $\omega : E \to \mathbb{R}$ (asuma que todos los costos son distintos). Sea $e^\star$ un arco dado (fijo) de G.

(i).- (1.5 pts) Pruebe que $e^\star$ está en todo árbol generador de peso mínimo de G si para algún $S \subseteq V(G)$ se tiene que

$$\omega(e^\star) = \min_{e \in \delta(S)} \omega(e) \,.$$

(ii).- (1.5 pts) Pruebe que $e^\star$ no está en ningún árbol generador de peso mínimo de G si pertenece a algún circuito $C \subseteq E(G)$ tal que

$$\omega(e^\star) = \max_{e \in C} \omega(e) \,.$$

(iii).- (1.5 pts) Pruebe que $e^* = uv$ no pertenece a ningún árbol generador de peso mínimo de G si y sólo si existe un (v, u)–camino $\mathcal{P}$ en G tal que $\omega(e^*) > \omega(e)$ para todo $e \in \mathcal{P}$.

(iv).- (1.5 pts) Muestre que en tiempo $O(|V| + |E|)$ se puede decidir si e^* pertenece a algún árbol generador de peso mínimo de G.

PROBLEMA 3: Sea $G = (V, E)$ un digrafo con función de costos en los arcos $\omega : E \to \mathbb{R}$ (pudiendo tomar valores negativos). Para $h : V \to \mathbb{R}$ se define la función de costos $\widehat{\omega} : E \to \mathbb{R}$ por

$$\widehat{\omega}(uv) = \omega(uv) + h(u) - h(v) \,.$$

(i).- (2.0 pts) Muestre que si $\mathcal{P}$ es un dicamino de costo mínimo en G con respecto a ω, entonces también es un dicamino de costo mínimo con respecto a $\widehat{\omega}$, y que G posee dicircuitos de costo negativos con respecto a ω si y sólo si tiene dicircuitos de costo negativo con respecto a $\widehat{\omega}$.

(ii).- (2.0 pts) Sea $G' = (V', E')$ tal que $V' = V \cup \{s\}$, $s \notin V$, y $E' = E \cup \{sv : v \in V\}$. Sea además $\omega' : E' \to \mathbb{R}$ tal que ω' es igual a ω sobre E y toma el valor 0 sobre $E' \setminus E$. Si G no posee dicircuitos de costo negativo, entonces en G' esta bien definido $\delta'(s, v)$ — el costo de un (s, v)–dicamino de costo mínimo. Muestre que si $h(v) = \delta'(s, v)$, entonces la función de costo $\widehat{\omega}$ definida en la parte anterior toma valores no–negativos.

(iii).- (2.0 pts) Sea $T(n, m)$ el tiempo que toma Dijkstra en entradas correspondientes a n nodos y m arcos. Se sabe que si Dijkstra se implementa en base a heaps de Fibonacci, entonces $T(n, m) = O(n \log n + m)$. Proponga un algoritmo (describalo mediante pseudocódigo) que en la entrada G tal que $|E| = o(|V|^2)$, determine en tiempo $o(|V|^3)$ (asintóticamente más rápido que el algoritmo de Floyd–Warshall), el costo de un (u, v)–dicamino de costo mínimo en G para cada par de nodos u y v de G.