MA3705 - Algoritmos Combinatoriales
Profesora: Maya Stein
Auxiliar: Tomás Martínez M.
Fecha Auxiliar: 11 de Octubre, 2018

Tarea 4: Caminos Mínimos

Importante: Cuentan como parte de la tarea sólo las preguntas P1 a) y b) y la P2, el resto está para que ejerciten, se entretengan o para que las muestre el auxiliar.
La P1 a) y b) deben ser entregadas por escrito.

P1. Camino mínimo con pesos enteros

Sea $G = (V, E)$ un grafo dirigido y $w : E \to \mathbb{N}$ una función de peso, considere $W = \max_e w(e)$. Sea k el número de dígitos de W escrito en binario, para $i \in [k]$ definimos $w_i(e) = \lfloor \frac{w(e)}{2^{k-i}} \rfloor$, o en palabras, w_i considera solo los i dígitos más significativos de w. Definimos también $d_i(u, v)$ como el peso de un uv-camino mínimo para w_i. Nuestro objetivo es mostrar un algoritmo que dado un vértice s calcule $d(s, v)$ para todo $v \in V$ en tiempo $O(kE) = O(Elog(W))$. (Supondremos $|E| \geq |V| - 1$).

a) Muestre que es posible calcular $d_1(s, v)$ para todo $v \in V$ en tiempo $O(E)$.

b) Suponga que $d(s, v) \leq E$, pruebe modificando la cola de prioridad de Djikstra que es posible calcular $d(s, v)$ para todo $v \in V$ en tiempo $O(E)$.

c) Nos interesa calcular d_i a partir de d_{i-1} y para eso probaremos un par de cosas, para empezar pruebe que:

$$2d_{i-1}(s, v) \leq d_i(s, v) \leq 2d_{i-1}(s, v) + |V| - 1$$

d) Para $i \in \{2, ..., k\}$ y $u, v \in V$ definimos:

$$\hat{w}_i(u, v) = w_i(u, v) + 2d_{i-1}(s, u) - 2d_{i-1}(s, v)$$

Pruebe que $\hat{w}_i(u, v)$ es un entero no negativo.

e) Sea $\hat{d}_i$ la distancia mínima a pesos $\hat{w}_i$. Pruebe que $d_i(s, v) = \hat{d}_i(s, v) + 2d_{i-1}(s, v)$.

f) Entregue el algoritmo deseado.

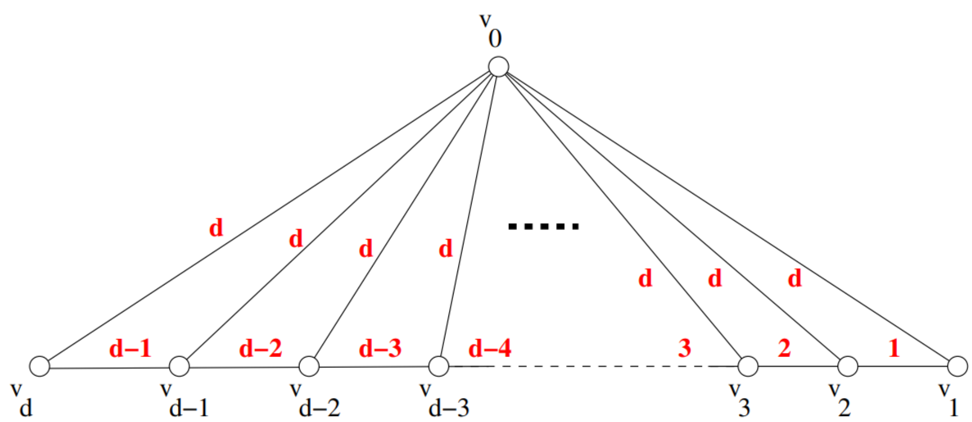
P2. Consideremos el siguiente grafo G con $d + 1$ nodos $V = \{v_0, v_1, ..., v_d\}$ tal que para cada $1 \leq i \leq d$ hay una arista de peso d entre v_0 y v_i, y para cada $1 \leq i < d$

a) Aplique el algoritmo de Kruskal en G. Explique como procede y cual es el MST que obtiene. Muestre d diferentes arboles de peso mínimo de G.

b) Sea T un ST de G y asumamos que existen $1 \leq i \leq j \leq d$ tal que $\{v_0, v_i\}, \{v_0, v_j\} \in E(T)$ y, para cada $i < k < j$ $\{v_0, v_i\} \notin E(T)$. Muestre que T no es MST.

c) Myestre que hay exactamente d MST distintos en G.

d) Muestre que ningún MST es un shortest path tree (Esto es mostrar que para cualquier MST T de G y cualquier $v \in V$, T no es un shortest-path tree con raíz en v).



P3.

Sea G $= (V, E)$ un grafo dirigido con función de peso $w : E \to \mathbb{R}$. Denotaremos el peso promedio de un ciclo C como $\mu(C)$. El objetivo de este problema es dar con un algoritmo que permita calcular $\mu^* = \min_{C ciclo} \mu(C)$. Asumiremos (sin pérdida de generalidad) que todo nodo es alcanzable desde un nodo s.
Denotaremos por $d(s, v)$ al peso de un sv-paseo mínimo y $d_k(s; v)$ será el peso de un sv-paseo mínimo con exactamente k arcos (si no hay un un paseo de s a v con exactamente k arcos, entonces $d_k(s, v) = \infty$).

a) Muestre que si $\mu^* = 0$ entonces pasan las siguientes cosas

   i) Para todo v en V $d(s, v) = \min_{0 \le k < n} d_k(s, v)$.

   ii) Para todo v en V $\max_{0 \le k < n}[d_n(s, v) - d_k(s, v)] \ge 0$.

   iii) Si u, v son vértices cualquiera de un ciclo C de peso promedio mínimo, y x es el peso del uv-camino contenido en C, entonces
   $$d(s, v) = d(s, u) + x$$

   iv) Si C es un ciclo de peso promedio mínimo, entonces existe $v \in C$ tal que V $\max_{0 \le k < n}[d_n(s, v) - d_k(s, v)] = 0$

b) Estudie el efecto que tiene sobre μ^* y sobre $\max_{0 \le k < n} \frac{[d_n(s,v) - d_k(s,v)]}{n-k}$ el incrementar en una constante t el peso de cada arco de G. Concluya que

$$\mu^* = \min_{v \in V} \max_{0 \le k < n} \frac{[d_n(s, v) - d_k(s, v)]}{n - k}$$

c) De un algoritmo que calcule μ^* en tiempo $O(nm)$.