MA3705. Algoritmos Combinatoriales 2014.
Profesor: José Soto

Problemas Controlables (parte 2).

Problema 7. Sea $G = (V, E)$ un grafo no dirigido. Un conjunto $W \subseteq V$ de vértices se dice *independiente* si el grafo inducido $G[W]$ no posee aristas. El tamaño máximo de un conjunto independiente de G se denota por $\alpha(G)$.

a) Diseñe un algoritmo basado en programación dinámica para encontrar $\alpha(G)$ para $G = (V, E)$ **árbol** ¿Puede lograr que su algoritmo corra en tiempo $O(n)$?

   Indicaciones: Para simplificar el problema primero oriente G de modo que sea una arborescencia con raíz $s \in V$. Para cada nodo $v \in V$ defina G_v como la subarborescencia de G de raíz v. ¿Qué relación existe entre $\alpha(G_v)$, $\alpha(G_v - v)$ y los correspondientes valores para los "hijos" de v en G?

b) Modifique su algoritmo para retornar un conjunto independiente W de tamaño $\alpha(G)$ con la misma complejidad asintótica.

Problema 8.

a) Dado un vector $v \in \mathbb{R}^n$, diseñe un programa dinámico que determine el subvector $v[i, j] = (v_i, v_{i+1}, \ldots, v_j)$ de mayor suma. ¿Cuál es la complejidad de su algoritmo? (Su algoritmo debe ser más eficiente que el algoritmo trivial que calcula la suma de todos los $v[i, j]$ de manera independiente que toma tiempo $O(n^3)$).

b) Dada una secuencia $S = v_1, v_2, \ldots, v_n$ de enteros en el rango $\{0, 1, \ldots, k\}$ diseñe un programa dinámico que particione S en dos subconjuntos S_1 y S_2 con sumas parciales $s_1 = \sum_{v \in S_1} v$ y $s_2 = \sum_{v \in S_2} v$ tal que $|s_1 - s_2|$ sea mínimo. Su algoritmo debe ser polinomial en n y lineal en k.
   Ayuda: Inspírese en la secuencia de preguntas: ¿Posee S un subconjunto de suma j?

Problema 9. Deseamos resolver un sistema S de ecuaciones con n variables x_i, y m ecuaciones del tipo $x_i - x_j \leq c_{i,j}$, donde $c_{i,j} \in \mathbb{R}$ es constante. Construya el digrafo $G = (V, E)$ con n nodos v_i asociados a las variables x_i y m arcos (i, j) asociados a la ecuacion de constante $c_{i,j}$. Asigne además a cada arco (i, j), largo $c(i, j)$.

a) Demuestre que S no tiene solución si y solo si G tiene un ciclo de costo negativo.

b) Diseñe un algoritmo para encontrar una solución de S si esta existe, y que además reporte si no hay solución.
   Ayuda: Agregue un nodo extra v_0 a G que denote el valor 0, y arcos desde v_0 a los otros nodos con pesos adecuados. Interprete el problema como uno de caminos de largo mínimo.

Problema 10. [Algoritmo de Strassen para multiplicar matrices.] Dadas dos matrices $A \in \mathbb{Z}^{n \times n}$ y $B \in \mathbb{Z}^{n \times n}$, calcular la matriz AB se puede hacer trivialmente en $O(n^3)$ multiplicaciones y sumas. En esta pregunta mostraremos una técnica para calcular el producto de A y B, en el caso $a = b = c =: n = 2^t$ usando una cantidad subcúbica de multiplicaciones y sumas (comentario: Si los enteros involucrados están acotados por una cantidad fija, entonces este algoritmo es verdaderamente fuertemente polinomial y corre en tiempo $o(n^3)$). Sea $t \geq 0$, $n = 2^t$ y $A, B \in \mathbb{Z}^{n \times n}$, note que podemos escribir

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}, B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix},$$

donde $A_{i,j}, B_{i,j} \in \mathbb{Z}^{n/2\times n/2}$. Definimos las siguientes matrices auxiliares

$$\begin{aligned}
Q_1 &= (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2})\\
Q_2 &= (A_{2,1} + A_{2,2})B_{1,1}\\
Q_3 &= A_{1,1}(B_{1,2} - B_{2,2})\\
Q_4 &= A_{2,2}(-B_{1,1} + B_{2,1})\\
Q_5 &= (A_{1,1} + A_{1,2})B_{2,2}\\
Q_6 &= (-A_{1,1} + A_{2,1})(B_{1,1} + B_{1,2})\\
Q_7 &= (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2}).
\end{aligned}$$

a) Demuestre que $AB = \begin{pmatrix} Q_1 + Q_4 - Q_5 + Q_7 & Q_3 + Q_5 \\ Q_2 + Q_4 & Q_1 + Q_3 - Q_2 + Q_6 \end{pmatrix}$.

b) Llame $f(t)$ al número de multiplicaciones escalares necesarias para multiplicar dos matrices de dimensiones $2^t \times 2^t$ mediante la fórmula calculada anteriormente. Calcule $f(0)$ y encuentre una recursión para f de la forma $f(t) = 7f(t-1) + dt^2$.

c) Pruebe que $f(t) = O(7^t)$ y deduzca que el algoritmo resuelve el problema usando $O(n^{\log_2 7}) \approx O(n^{2,807})$ operaciones.

d) Suponga que G es un grafo simple. Es fácil crear un algoritmo que determine si G contiene un triángulo en $O(n^3)$ o incluso en $O(nm)$ operaciones (inténtelo). Usando las partes anteriores, diseñe un algoritmo para determinar si G contiene un triángulo en tiempo $O(n^{\log_2 7})$. (**Ayuda:** Suponga primero que G tiene exactamente 2^t vértices).

Problema 11.

a) Sea $G = (V, E)$ un grafo simple **no** dirigido y no acíclico, y $\ell \colon E \to \mathbb{R}_+$ una funcion de largos no negativos. Diseñe un algoritmo para encontrar el ciclo de G de largo minimo. ¿Cuán rápido puede lograr que sea su algoritmo? Tenga cuidado: Simplemente reemplazar aristas por arcos dirigidos genera ciclos dirigidos de 2 arcos que NO corresponden a aristas en el grafo original.

b) La cintura de un grafo G simple y no dirigido se define como el mínimo número de vértices en un ciclo de G (o infinito, si G es acíclico). Diseñe un algoritmo para encontrar la cintura de G. Su algoritmo debe ser más rápido que el usado en la parte anterior.

Problema 12. Una agencia de viaje desea crear un software para asignar vuelos a sus clientes. Posee una base de datos con los siguientes conjuntos

- A es el conjunto de aeropuertos.
- F es el conjunto de los vuelos posibles. Cada vuelo es una tupla $f = (a, b, \tau, \tau')$ donde a y b son los aeropuertos de origen y destino respectivamente, y τ y τ' son los tiempos de partida y llegada respectivamente (son números reales con $\tau < \tau'$). Notar que pueden haber varios vuelos distintos entre dos aeropuertos.

Una **ruta** del aeropuerto a al aeropuerto b es una secuencia de vuelos que una persona podría usar para llegar de a a b (note que en particular, el aeropuerto de llegada de un vuelo debe ser el de partida del siguiente, y los tiempos deben ser consistentes)

Dados A, F un aeropuerto de origen s y un tiempo de partida τ_s, su labor es diseñar un algoritmo que calcule para cada destino $v \in A$, la ruta desde s a v que llega lo más temprano posible. ¿Cuál es la complejidad de su algoritmo?

Indicación: Transforme el problema en uno de caminos de largo mínimo.

Problema 13.

a) Demuestre el Teorema de König directamente usando el Teorema de matrimonio de Hall.

b) Pruebe que el Teorema de matrimonio de Hall también vale para multigrafos bipartitos.

c) Sea $G = (L, R; E)$ un multigrafo d-regular, es decir, donde cada vértice es incidente a d aristas. Pruebe que G se tiene un matching perfecto, y de hecho G se puede particionar en d matchings perfectos.

d) Usted posee un naipe inglés estándar de 52 cartas (13 números y 4 pintas). Muestre que no importa como particione el mazo en 13 pilas P_i de 4 cartas cada una, siempre es posible encontrar una carta x_i de cada pila P_i de modo que $\{x_1, x_2, \dots, x_{13}\}$ contenga una carta de cada número (es decir, 1 as, 1 dos, 1 tres, etc.)

Problema 14.

a) Sea $G = (L, R; E)$ un grafo bipartito y $k \in \mathbb{N}$. Generalice el teorema de Hall del siguiente modo. Muestre que si para cada $X \subseteq L$, $|N(X)| \geq |X| - k$ entonces existe un matching que cubre $|L| - k$ elementos de L

b) Sea $G = (L, R; E)$ un grafo bipartito y $k \in \mathbb{Z}^+$. Generalice el teorema de Hall del siguiente modo. Muestre que si para cada $X \subseteq L$, $|N(X)| \geq k|X|$ entonces podemos encontrar una asignación $\varphi\colon L \to \binom{R}{k}$ tal que todos los conjuntos $\{\varphi(i)\colon i \in L\}$ son disjuntos.

c) Muestre como reducir el problema de encontrar un matching de peso máximo en un grafo bipartito $G = (L, R; E)$ no necesariamente completo ni balanceado (i.e. $|L|$ y $|R|$ no necesariamente iguales) a un problema de asignación en un grafo auxiliar. Si necesita usar costos arbitrariamente grandes o pequeños en los arcos indique un valor preciso que funcione (no use infinito, sino que indique un valor adecuado en función de los parámetros del grafo original).

Problema 15. Sea $G = (V, E)$ un grafo dirigido simple y completo (es decir, $E = V \times V$) y $w\colon E \to \mathbb{R}$ una función de peso cualquiera. Un cubrimiento (exacto) por ciclos de G es una colección de ciclos dirigidos $\mathcal{C}$ de G, tal que cada vértice $v \in V$ aparece exactamente un ciclo $C \in \mathcal{C}$. Definimos además el peso de $\mathcal{C}$ como

$$\sum_{C \in \mathcal{C}} w(C) = \sum_{C \in \mathcal{C}} \sum_{e \in E(C)} w(e).$$

Diseñe un algoritmo polinomial que encuentre un cubrimiento por ciclos de G de peso mínimo.
Indicación: Reduzca el problema anterior a un problema de asignación en un grafo bipartito.

Problema 16. Sea $G = (V, E)$ un grafo bipartito y considere los polítopos

$$P(G) = \Big\{x \in \mathbb{R}^E \colon x \geq 0 \text{ y } \sum_{e \in \delta(v)} x_e \leq 1 \, \forall v \in V\Big\} \;\text{ y }\; P_{\text{matching}}(G) = \operatorname{conv}(\{\chi_M \colon M \text{ matching en } G\}).$$

Demuestre que $P(G) = P_{\text{matching}}$.

Indicación: Para probar la inclusión no trivial, considere la siguiente construcción auxiliar. Sean $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ dos copias disjuntas de G con $V_i = \{v_i \colon v \in V\}$, $i \in \{1, 2\}$. Defina un nuevo grafo H con $V(H) = V_1 \cup V_2$ y $E(H) = E_1 \cup E_2 \cup \{(v_1, v_2) \colon v \in V\}$. Para probar la propiedad, encuentre una relación entre los matchings de G y los matchings perfectos de H; y una relación similar entre los vectores de $P(G)$ y los vectores de $P_{\text{match. perfecto}}(H)$, cuya caracterización se vió en clases.