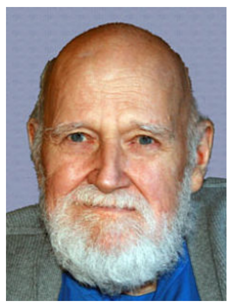
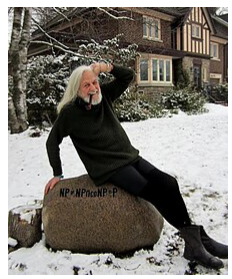
Flujo

Técnincas de Diseño de Algoritmos, 2C 2024 TM

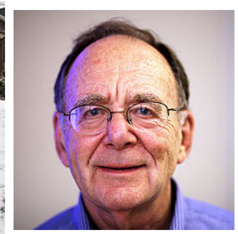
(a) Delbert Ray Fulkerson



(b) Lester Randolph Ford jr.



(c) Jack Edmonds



(d) Richard Manning Karp

Enunciados

1 La gira de Pablo

1.1

La banda de "*rock*" de Pablo y Ramón ya está planeando sus años de éxito con múltiples giras nacionales. Su objetivo es planear muchas giras que terminen todas en Ushuaia, tocando en el medio por bares en todas las ciudades que les queden de camino. Tienen un mapa de los lugares donde podrían tocar, unidos por rutas unidireccionales. Pero Pablo tiene miedo de que después de viajar por una ruta, los paparazzis se enteren de que pasan por ahí y los molesten la próxima vez que viajen, así que le interesa planear sus giras de forma de no usar nunca una ruta dos veces. ¿Cuántas giras distintas puede planear?

1.2

Ramón se queja de que la restricción de terminar en Ushuaia no tiene sentido y que solo lo estaban haciendo porque a Pablo le gusta la que manejaba el bar. Propone una nueva restricción: Pueden terminar en cualquier capital provincial, así se pueden tomar un avión de vuelta más fácil. ¿Cuántas giras distintas pueden hacer ahora?

1.3

Ceci, la manager de la banda, les informa que algunas de las ciudades intermedias donde piensan pasar son lugares *de buen gusto*, y, por la calidad de su música, una vez que tocan ahí no van volver a ser bienvenidos. Ahora la banda puede pasar a lo sumo una vez por cada ciudad de buen gusto.

2 El simposio de Martín

Martín está organizando un simposio sobre teoría de grafos y sabe que la gente no viene si no hay buena comida. Tiene un conjunto P de platos que puede cocinar. Hizo una encuesta entre los investigadores y formuló para cada uno una lista $I_i \subseteq P$ de sus platos favoritos. Sabe que para cada plato $p_j \in P$ puede cocinar a lo sumo q_j unidades. Martín quiere saber cuál es la máxima cantidad de investigadores a los que le va a poder servir su comida favorita.

3 Tuki en GH

Tuki fue uno de los afortunados en entrar a la casa de GH (Grupo Humano). En este juego cada semana los jugadores votan por la eliminación de otro jugador. Cada jugador vota a un único jugador y los jugadores con más votos son enviados a un desafío.

Tuki sabe que conviene mantener la votación lo más pareja posible, para evitar problemas en la casa. El sabe que:

- Hay J jugadores y K grupos.
- El jugador i pertenece al grupo g_i.
- El jugador i está dispuesto a votar al conjunto N_i de jugadores.
- Si algún grupo G_l recibe más votos que el doble de la cantidad de jugadores en el grupo, entonces se va a sentir atacado.
- Si un jugador individual i recibe más de 3 votos, entonces también se siente atacado.

Quiere decidir, dados los J jugadores junto a sus conjuntos N_i y la descripción de los K grupos, si es posible que la votación logre que ningún grupo ni jugador se sienta atacado.

4 Victoria y las charlas

Ya confirmada la lista de asistentes al simposio, a Victoria le tocó la tarea de distribuirlos entre todas las charlas que se van a dar al mismo tiempo. Como cada una de las C charlas tiene un cupo máximo de A_i personas, cada asistente le informó su lista I_j de charlas que le interesan. Como Victoria conoce mucho el chisme de la comunidad de investigación de teoría de grafos, ella sabe que todos los investigadores se pueden dividir en R riñas de por lo menos dos investigadores cada una. Sabemos que cada investigador pertenece a exáctamente una riña.

Nos interesa saber si hay alguna forma de organizar a todos los participantes en las distintas charlas de forma que se respeten sus deseos y nadie se pelee.

5 Torres

En una cómoda tarde de primavera, Lampone y Santos se encontraban jugando no tan apaciblemente al ajedrez. Tras su tercer derrota, Lampone arrojó el tablero de mármol por los aires, logrando como resultado que tras la caída varios casilleros del mismo se resquebrajaran.

Santos, en vez de enojarse, observó con curiosidad que ahora, en este tablero con casilleros "tapados", era posible colocar más de una torre por fila sin que estas se amenacen, si se interpreta que dos torres se amenazan mutuamente si y solamente si se encuentran en la misma fila (o columna) y no hay un casillero roto entre ellos. Interesado en esta cualidad se propuso encontrar la máxima cantidad de torres que se pueden colocar en este tablero sin que se amenacen mutuamente.

Ejercicio bonus fuerte

En un hospital hay k períodos disjuntos de vacaciones (por ej. Semana Santa, Feriado puente de Güemes, etc.). Cada período consiste de una cierta cantidad de días contiguos de vacaciones. Conocemos cada conjunto $D_j = \{d_{j1}, \cdots, d_{jr}\}$ de días contiguos que conforman el período j. En este hospital hay n medicxs y cada cual tiene un conjunto Si de días disponibles para trabajar durante los períodos de vacaciones. Por ejemplo, una médica puede tener disponible viernes y sábado de semana santa y el lunes del feriado de Güemes. Queremos encontrar una asignación que cumpla:

- Nadie tiene asignado más que C días totales para trabajar en vacaciones (y solo dentro de sus días disponibles).
- Cada día de vacaciones tiene asignada una única persona para trabajar ese día.
- Para cada período j unx medicx sólo puede tener, como máximo, un día asignado dentro de D_j . Es decir, quizá tiene disponibles jueves, viernes,

sábado y domingo de semana santa pero solo se le puede asignar uno de esos días

Teoría, estrategias, y tips

Flujo válido

Dado una red de flujo $G = \langle V, E, c \rangle$, para que $f : E \to \mathbb{N}$ sea un flujo válido tiene que cumplir:

$$\forall e \in E,\ 0 \leq f(e) \leq c(e)$$

$$\forall v \in V - \{s, t\}, \sum_{u \in N_{in}(v)} f(u \to v) = \sum_{u \in N_{out}(v)} f(v \to u)$$

Dado una función de flujo válido, su flujo es:

$$|f| = \sum_{u \in N_{out}(s)} f(s \to u) = \sum_{u \in N_{in}(t)} f(u \to t)$$

Algoritmo FF/EK

El algoritmo devuelve una función de flujo válida en $O(min\{|E| \cdot F, |V| \cdot |E|^2\})$, donde F es el flujo máximo de la red. Podemos también acotar a F por $|V| \cdot U$, donde U es la máxima capacidad de una arista.

Correctitud de nuestro modelo

Tenemos el mundo de las soluciones a nuestro problema abstracto y el mundo de resultados en nuestro modelo. Nos gustaría demostrar que la solución óptima a nuestro problema es equivalente a la solución óptima en nuestro modelo.

Una forma de mostrar eso es demostrar que los conjuntos de soluciones válidas son iguales. Es decir, si A es el conjunto de soluciones válidas al problema y B las soluciones válidas en nuestro modelo, queremos ver que $A = B$, así trivialmente el óptimo va a ser igual en ambos lados.

En flujo suele ser cómodo demostrar esto separando la ida y la vuelta.

Para mostrar que $B \subseteq A$, queremos, para cada cada solución válida f al modelo, tener una solución equivalente a nuestro problema. Por ejemplo, si lo que buscábamos era la mayor cantidad de caminos disjuntos en mapa, tenemos que mostrar que dado un flujo f en nuestro modelo hay un conjunto de $|f|$ caminos disjuntos en nuestro problema.

Similarmente, para mostrar que $A \subseteq B$, tenemos que probar que para cada solución a nuestro problema podemos generar un flujo f de optimalidad equivalente.

Más de un sumidero

Las redes de flujo solo admiten una fuente y un sumidero. Si tenemos un conjunto Q de vértices que queremos que actúen como sumideros, lo que podemos hacer es crear un nuevo vértice t que actúe como sumidero y conectar cada $q \in Q$ a t con $c(q \to t) = \infty$.

Limitar la cantidad de flujo que pasa por un vértice

Si queremos que por un vértice v no pueda pasar más de l unidades de flujo, podemos separar a v en v_1 y v_2, donde:

- $N_{in}(v_1) = N_{in}(v)$
- $N_{out}(v_1) = \{v_2\}$
- $N_{out}(v_2) = N_{out}(v)$
- $N_{in}(v_2) = \{v_1\}$
- $c(v_1 \to v_2) = l$.

Unificar capas

Puede pasar a veces que no sea suficiente con las entidades concretas del problema como capas para modelar las restricciones, sino que tenemos que hacer alguna combinación entre ellas, o con entidades abstractas.