

# TEORÍA DE ALGORITMOS 1

## Guía de ejercicios: Redes de Flujo

Por: ING. VÍCTOR DANIEL PODBEREZSKI  
vpodberezski@fi.uba.ar

---

### Enunciados

Para cada ejercicio considere, explicar el proceso de creación de la red de flujo, el proceso de resolver el problema mediante la red de flujo y la interpretación del resultado. Presentar pseudocódigo y realizar análisis de complejidad. Brinde uno o varios ejemplos representativos del problema.

### Referencias

- ★ Fácil
- ★★ Medio
- ★★★ Difícil

1. ★ La sala de guardia de un hospital tiene que tener al menos un médico en todos los feriados y en los fines de semana largos de feriados. Cada profesional indica sus posibilidades: por ejemplo alguien puede estar de guardia en cualquier momento del fin de semana largo del 9 de julio (p. ej. disponibilidad de A para el 9 de julio = (Jueves 9/7, Viernes 10/7, Sábado 11/7, Domingo 12/7)), también puede suceder que alguien pueda sólo en parte (por ejemplo, disponibilidad de B para 9 de julio = (Jueves 9/7, Sábado 11/7, Domingo 12/7)). Aunque los profesionales tengan múltiples posibilidades, a cada uno se lo puede convocar para un solo día (se puede disponer de B sólo en uno de los tres días que indicó). Para ayudar a la sala de guardia a planificar cómo se cubren los feriados durante todo el año debemos resolver el problema de las guardias: Existen  $k$  períodos de feriados (por ejemplo, 9 de julio es un período de jueves 9/7 a domingo 12/7, en 2019 Día del Trabajador fue un período de 1 día: miércoles 1 de mayo, etc.).  $D_j$  es el conjunto de fechas que se incluyen en el período de feriado  $j$ -ésimo. Todos los días feriados son los que resultan de la unión de todos los  $D_j$ . Hay  $n$  médicos y cada médico  $i$  tiene asociado un conjunto  $S_i$  de días feriados en los que puede trabajar (por ejemplo B tiene asociado los días Jueves 9/7, Sábado 11/7, Domingo 12/7, entre otros).  
Proponer un algoritmo polinomial (usando flujo en redes) que toma esta información y

---

devuelve qué profesional se asigna a cada día feriado (o informa que no es posible resolver el problema) sujeto a las restricciones:

- Ningún profesional trabajará más de  $F$  días feriados ( $F$  es un dato), y sólo en días en los que haya informado su disponibilidad.
  - A ningún profesional se le asignará más de un feriado dentro de cada período  $D_j$ .
2. ★ La red de transporte intergaláctico es una de las maravillas del nuevo imperio terráqueo. Cada tramo de rutas galácticas tiene una capacidad infinita de transporte entre ciertos planetas. No obstante, por burocracia - que es algo que no los enorgullece - existen puestos de control en cada planeta que reduce cuantas naves espaciales pueden pasar por día por ella. Por una catástrofe en el planeta  $X$ , la tierra debe enviar la mayor cantidad posible de naves de ayuda. Por un arreglo, durante un día los planetas solo procesaran en los puestos de control aquellas naves enviadas para esta misión. Tenemos que determinar cuál es la cantidad máxima de naves que podemos enviar desde la tierra hasta el planeta  $X$ . Sugerencia: considerar a este un problema de flujo con capacidad en nodos y no en ejes
3. ★ La compañía eléctrica de un país nos contrata para que le ayudemos a ver si su red de transporte desde su nueva generadora hidroeléctrica hasta su ciudad capital es robusta. Nos otorgan un plano con la red eléctrica completa: todas las subestaciones de distribución y red de cableados de alta tensión. Lo que quieren que le digamos es: cuantas secciones de su red se pueden interrumpir antes que la ciudad capital deje de recibir la producción de la generadora? (Sugerencia: investigue sobre el Teorema de Menger) Puede informar cual es el subconjunto de ejes cuya falla provoca este problema?
4. ★★ Contamos con una red de Flujo definida sobre el grafo  $G=(V,E)$  y tenemos una asignación de flujo  $f(e)$  sobre  $G$ . Nos solicitan elaborar un algoritmo que en base a esta información nos indique cómo se actualiza el flujo si uno de los ejes tiene un cambio de capacidad (puede ser positiva o negativa). Debemos evitar volver a ejecutar el algoritmo de Ford-Fulkerson desde cero. Explicar los diferentes casos que podrían suceder. Utilizar los conceptos de flujo máximo/corte mínimo en su explicación. Brindar pseudocódigo de su propuesta y análisis de complejidad.
5. ★★ La red ARPANET, antecesora de internet, se creó para seguir funcionando incluso ante fallas en parte de su red. El país "Atrasoñia" - que se mantuvo cerrado a los avances tecnológicos de las últimas décadas - ha decidido construir su propia red de redes. Han leído la documentación desclasificada de ARPANET y se han instruido en conectividad de redes. Proponen una red informática para unir sus principales organismos estatales. Nos convocan para que validemos su diseño. Debemos responder: ¿Cuántos cables de datos de la red se tienen que romper antes que la conectividad del grafo se rompa? (tener en cuenta que los cables de datos son bidireccionales) ¿Cuántos nodos se tienen que romper antes que el grafo restante deje de ser conexo? (Sugerencia: piense en transformar de alguna forma los nodos para resolverlo mediante lo creado en el punto a)
6. ★★ Para un evento solidario un conjunto de  $n$  personas se ofrecieron a colaborar. En el evento hay  $m$  tareas a desarrollar. Cada tarea tiene un cupo máximo de personas que la puede realizar. A su vez cada persona tiene ciertas habilidades que la hacen adecuadas para un subconjunto de tareas. Proponga una solución mediante red de flujos que maximice la cantidad de personas asignadas a las tareas. ¿Hay forma de lograr asegurarnos un piso

---

mínimo de personas en cada tarea? ¿Cómo impacta en la solución presentada en el punto anterior?

7. ★ La policía de la ciudad tiene “ $n$ ” comisarías dispersas por la ciudad. Para un evento deportivo internacional deben asignar la custodia de “ $m$ ” centros de actividades. Una comisaría y un centro de actividades pueden ser emparejados si y sólo si la distancia entre ellos no es mayor a un valor  $d$ . Contamos con la distancia entre todos los centros y las comisarías. Una comisaría sólo puede custodiar un centro. El centro puede ser custodiado por una comisaría. Determinar si es posible la asignación de tal forma que todos los centros estén custodiados. ¿Cómo modificaría la resolución del problema si en lugar de que cada centro de actividades  $i$  tenga que ser asignado a una sola comisaría, tenga que ser asignado a  $m_i$  comisarías? ¿Cómo modificaría la resolución del problema si además hubiera una restricción entre comisarías que implicaría que una comisaría  $N_i$  y una  $N_j$  no pudieran ser asignadas juntas a un centro  $M_i$ ? ¿Para qué casos dejaría de ser eficiente la resolución?
8. ★ Una compañía minera nos pide que la ayudemos a analizar su nueva explotación. Ha realizado el estudio de suelos de diferentes vetas y porciones del subsuelo. Con estos datos se ha construido una regionalización del mismo. Cada región cuenta con un costo de procesamiento y una ganancia por extracción de metales preciosos. (En algunos casos el costo supera al beneficio). Al ser un procesamiento en profundidad ciertas regiones requieren previamente procesar otras para acceder a ellas. La compañía nos solicita que le ayudemos a maximizar su ganancia, determinando cuales son las regiones que tiene que trabajar. Tener en cuenta que el costo y ganancia de cada región es un valor entero. Para cada región sabemos cuales son aquellas regiones que le preceden. Resolver el problema planteado utilizando una aproximación mediante flujo de redes.
9. ★★ Un taller metalúrgico cuenta con un conjunto de  $M$  controles numéricos computarizados (CNC). Cada uno de ellos puede realizar trabajos de duración en bloques de 1 hora. Por otro lado, cuenta con  $N$  tareas a realizar. Cada tarea  $i$  tiene una duración de horas de desarrollo, una hora de posible comienzo (cuando puede comenzar a realizarse) y una hora de entrega (momento donde debe estar finalizada). Las tareas pueden realizarse parcialmente y utilizarse para su elaboración una o varias máquinas. Por ejemplo si la tarea A requiere 4 horas. Podría realizarse 1 hora en la máquina 1 y luego de un intervalo concluir su desarrollo en la máquina B (3 horas restantes). El jefe de planta nos solicita nuestra ayuda para que le ayudemos a determinar si podrá cumplir con la finalización de las tareas en tiempo y forma. Y en caso afirmativo le indiquemos cómo organizarla. Se solicita utilizando redes de flujos dar una solución al problema.
10. ★★ Una red de espías se encuentran diseminados por todo el país. Cada uno de ellos únicamente conoce a un número limitado de sus pares con los que puede tener contacto dejando un mensaje escrito en una ubicación conocida. Este conocimiento no es recíproco. En caso de una crisis la agencia puede enviar mensajes utilizando esta red desde su base principal a un determinado agente especial. Una cuestión importante es que una vez utilizado un espía para transmitir un mensaje durante el resto de la crisis no se vuelve a utilizar. La agencia desea conocer, dada su red y un agente de destino de sus mensajes. ¿Cuál es la mínima cantidad de espías que un rival podría neutralizar para reducir en un 30% la cantidad de mensajes máximos que puede enviar desde la base al agente? Utilizando redes de flujos dar una solución al problema.

- 
11. ★★ En un juego multijugador cooperativo de "p" participantes se muestra una grilla de  $n \times n$  celdas. Inicialmente en una posición al azar se colocan los avatares de los jugadores e igual cantidad de cuevas. Cada "gusano" se desplaza 1 celda por turno ocupando una circundante que esté vacía (como máximo tiene 4: arriba, abajo, izquierda y derecha). A medida que se desplaza crece y por lo tanto sigue ocupando por las que pasó anteriormente. El objetivo del juego es lograr que los "p" jugadores lleven a sus gusanos a las cuevas (es indiferente a cual pero cada uno tiene que estar en una cueva diferente). Nos solicitan que realicemos el pseudocódigo que verifique en un turno determinado si aún es posible que los jugadores ganen. Es decir que todos los gusanos puedan llegar a la cueva.
  12. ★ Se cuenta con una grafo  $G=(V,E)$  con capacidad 1 en cada uno de sus ejes. Existen 2 nodos que tomaremos como "s" fuente y "t" sumidero. Podemos determinar el flujo máximo F entre s-t. Se pide proponer un algoritmo eficiente que dado un valor "k", determine la cantidad mínima de ejes y cuáles de ellos eliminar para que el nuevo flujo máximo sea F-k. Determinar su complejidad y detallar mediante pseudocódigo y una explicación cómo funciona.
  13. ★ Una empresa de autobuses se conformó luego de la fusión de varias compañías menores. Actualmente tienen diferentes rutas que cubrir. Cada una con horario de inicio en una ciudad y finalización en otra. Existe la posibilidad de cubrir con un mismo micro diferentes rutas. Siempre la ruta comienza desde donde parte el micro, pero también puede pasar que el micro tenga tiempo suficiente para trasladarse hasta otro punto y cubrir otra ruta. Cuentan con una flota activa de N micros. Necesitan saber si les es posible cubrir con ella los requerimientos y si pueden contar con micros de backup ante la necesidad de controles programados de algunos de los móviles. Ayudar a resolver el problema mediante el uso de redes de flujo.
  14. ★ Un grupo de "n" amigos participan en un torneo de voley. En cada partido se debe presentar una planilla de "y" jugadores. Se juegan partidos 1 vez por semana, durante un periodo de 9 meses. Para lograr que juegue la mayoría se propuso que cada amigo no juegue más de 4 partidos. Esos partidos deben estar lo más separados posible, por lo que no pueden jugar más de 1 partido por mes. Finalmente se tiene que tener en cuenta que algunos amigos en ciertas fechas no pueden asistir por cuestiones personales. Proponer un algoritmo polinomial que utilizando esta información realice la asignación de jugadores por partido (o que indique que con esas restricciones no es posible realizarlo)
  15. ★ Demostrar que en una red de flujo pueden existir más de un corte mínimo. Brindar un algoritmo para determinar si dada una red de flujo tiene un único corte mínimo o no. Su complejidad temporal no debe ser mayor a la de hallar su flujo máximo.
  16. ★★ Demostrar que el flujo máximo en una red de flujo  $G=(V,E)$  puede siempre encontrarse como mucho en como máximo  $|E|$  caminos de aumento.

---

## Ejercicios resueltos

Cada ejercicio resuelto busca mostrar cómo se debe analizar, resolver y justificar la resolución del ejercicio, tanto en un trabajo práctico o en un parcial

1. ★ Definimos el Problema de la Evacuación de la siguiente manera: Se tiene un grafo dirigido  $G = (V, E)$  que describe una red de caminos. Tenemos una colección de nodos  $X \subset V$  que son los nodos poblados (ciudades) y otra colección de nodos  $S \subset V$  que son los nodos de refugio (supondremos que  $X$  y  $S$  son disjuntos). En caso de una emergencia queremos poder definir un conjunto de rutas de evacuación de los nodos poblados a los refugios. Un conjunto de rutas de evacuación es un conjunto de caminos en  $G$  tales que (i) cada nodo en  $X$  es el origen de un camino, (ii) el último nodo de cada camino es un refugio (está en  $S$ ), y (iii) los caminos no comparten aristas entre sí. Se pide: dados  $G$ ,  $X$  y  $S$ , mostrar cómo se puede decidir en tiempo polinomial si es posible construir un conjunto de rutas de evacuación (usar flujo en redes para eso. Construir la red adecuada).

### Pseudocódigo

Python

```
ford_fulkerson (G)
    Inicialmente el flujo  $f(e) = 0$  para todo eje  $e$  en el grafo  $G$ 
    Inicializar  $G_r$  grafo residual según  $G$  y flujo  $s$ - $t$   $f = 0$ 
    Mientras haya un camino  $s$ - $t$  en  $G_r$ 
        Sea  $P$  un camino  $s$ - $t$  en  $G_r$ 
        Sea  $w$  el valor de flujo del cuello de botella del camino  $P$ 

        Actualizar el flujo  $w$  en el camino  $P$  en  $G$ 
        Actualizar  $G_r$ 
    Retornar  $f$ 

# Aclaración: el pseudocódigo del armado de la red de flujo puede ser
# restando por una explicación
armado_red_flujo (G)
    # agregamos los nodos fuente y sumidero
    G.addVertex(F)
    G.addVertex(S)

    # conectamos las ciudades y refugios a la fuente y sumidero y definimos
    # las capacidades
    for  $v$  in  $G.X$ :
        G.addEdge(F, v)

    for  $v$  in  $G.S$ :
        G.addEdge(v, S)
```

---

```

        for e in G.edges:
            e.capacity = 1

armado_de_ruta (G, node, road)
    for v in node.adj:
        if G.edge(node, v).flow > 0:
            road.push(v)
            armado_de_ruta(G, v, road)

rutas_de_evacuación (G, X, S)
    armado_red_flujo(G) # modifica G in place
    flujo_maximo, G' = ford_fulkerson(G)
    if flujo_maximo != |X|:
        return "no es posible evacuar todas las ciudades"

    # Para determinar las rutas de evacuación se recorren los nodos del grafo
    # residual desde la fuente hasta el sumidero, seleccionando las rutas que
    # se forman

    node = F
    roads = []

    for v in F.adj:
        road = [F, v]
        armado_de_ruta(G, v, road)
        roads.push(road)

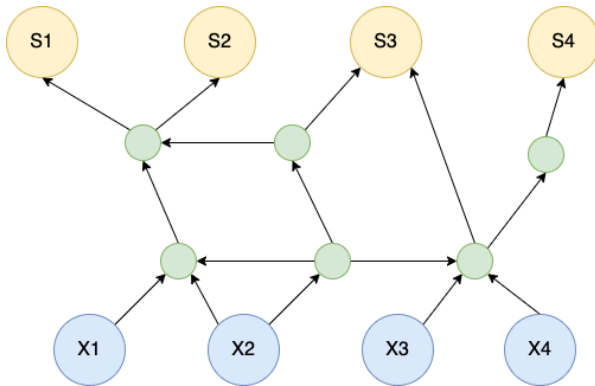
    return roads

```

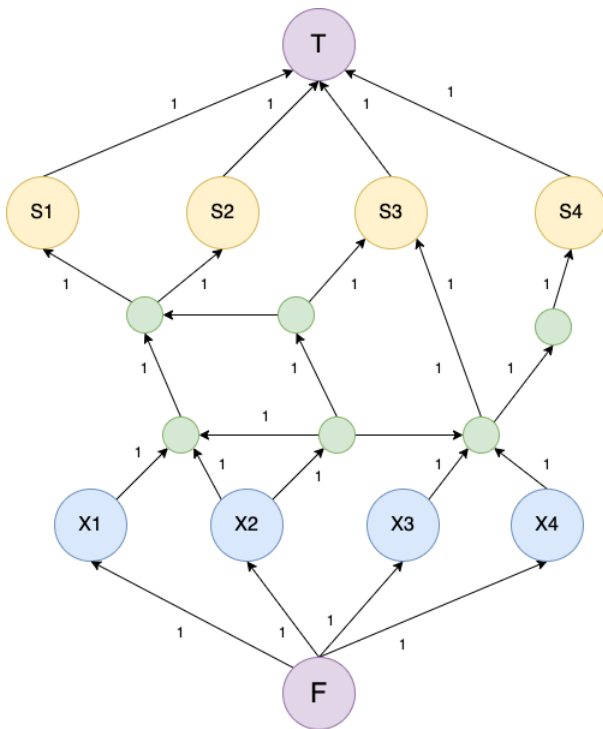
---

## Explicación de la solución

Red original



Red transformada



A la red original se le agregan los nodos Fuente F y Sumidero T. Se conecta la fuente a los nodos que representan ciudades y los refugios al sumidero T.

Los caminos no pueden compartir aristas. Por este mismo motivo todas las aristas intermedias tienen peso 1, para que solo 1 camino pueda pasar por allí desde  $X_i$  a  $S_j$ .

---

El resto de las aristas también tienen peso 1 para asegurarnos de que cada  $X_i$  llegue solo a un  $S_j$  y que cada  $S_j$  puede ser llegado por solo un  $X_i$ .

Ejecutando Ford Fulkerson verificamos que el flujo máximo sea igual a la cantidad de ciudades, lo cual representa que puede resolverse el problema.

Para determinar las rutas de evacuación se recorren los nodos del grafo residual desde la fuente hasta el sumidero, seleccionando las rutas que se forman.

### **Análisis de complejidad**

El armado de la red tiene una complejidad de  $O(V + E)$  dado que se deben agregar los nodos  $F$ ,  $T$ , conectarlas a los nodos ciudades y refugios y recorrer todas las aristas para asignarles capacidad 1.

Buscar el flujo máximo posee una complejidad de  $O(E * F_{\max}) = O(E * |X|)$ , dado que en el caso que exista una solución el flujo máximo posible es  $|X|$ .

Por último, el armado de las rutas posee una complejidad de  $O(V + E)$  (DFS)

La complejidad final resulta  $E * |X|$ . La misma es polinomial, pues depende de la cantidad de ciudades, parámetro del problema, y la cantidad de aristas.