4. REDES DE FLUJO

El problema del flujo máximo:

- ENTRADA: Una red N = (V, E, c, s, t,)
- PREGUNTA: Encuentrar un flujo de valor máximo en N.

El problema de corte mínimo:

- ENTRADA: Una red N = (V, E, c, s, t,)
- PREGUNTA: Encuentrar un corte (s, t) de capacidad mínima en N.

Si (S, T) es un corte mínimo:

- Si cambiamos la entrada agregando c > 0 a la capacidad de cada arista, entonces puede suceder que (S, T) ya no sea un corte mínimo (s, t).
- Si cambiamos la red multiplicando por c > 0 la capacidad de cada arista, la capacidad de cualquier corte (s, t) en la nueva red es c por su capacidad en la red original.

forward edges (aristas delanteras) -> si c (u, v) − f (u, v) > 0, cf (u, v) = c (u, v) − f (u,v) > 0
backward edges (aristas traseras) -> si f (u, v) > 0, cf (v, u) = f (u, v)
cf es la capacidad residual del grafo residual
aristas delanteras: queda capacidad para empujar más flujo a través de esta arista.
aristas traseras: hay unidades de flujo que se pueden redirigir a través de otros enlaces.
El máximo de los valor del flujo corresponde al mínimo de las capacidades del corte (S, T)
ALGORITMO 14: FORD FULKERSON

- Definición: El flujo devuelto por Ford-Fulkerson es el flujo máximo, algoritmo que calcula el flujo máximo.
- Coste: O(C(n + m)) = O(Cm), C -> max flow, n -> |V| and m -> |E|

ALGORITMO 15: EDMONDS KARP

- Definición: El flujo devuelto por Edmonds Karp es el flujo máximo, algoritmo que calcula el flujo máximo. (Elegir una buena ruta de aumento puede conducir a un algoritmo más rápido. Use BFS para encontrar una ruta de aumento en Gf.)
- Coste: O(m * n (n + m)) n -> |V| and m -> |E|

Circulación

Introducimos otro problema de flujo, para tratar con la oferta y la demanda dentro de una red. En lugar de tener un par de fuente/sumidero, el nuevo escenario considera un escenario de productor/consumidor. Algunos nodos no pueden producir una determinada cantidad de flujo.
D es la cantidad total de flujo adicional que debe transportarse desde las fuentes hasta los sumideros.
Hay una circulación para N = (V, E, c, d) si y sólo si el flow máximo en N′ tiene valor D.
Si todas las capacidades y demandas son números enteros, y existe una circulación, entonces existe una circulación de valor entero.
Existe un algoritmo de tiempo polinomial para resolver el problema de circulación.
Si todas las capacidades y demandas son números enteros, y existe una circulación, entonces podemos obtener una circulación de valor entero en el tiempo O(Dm).