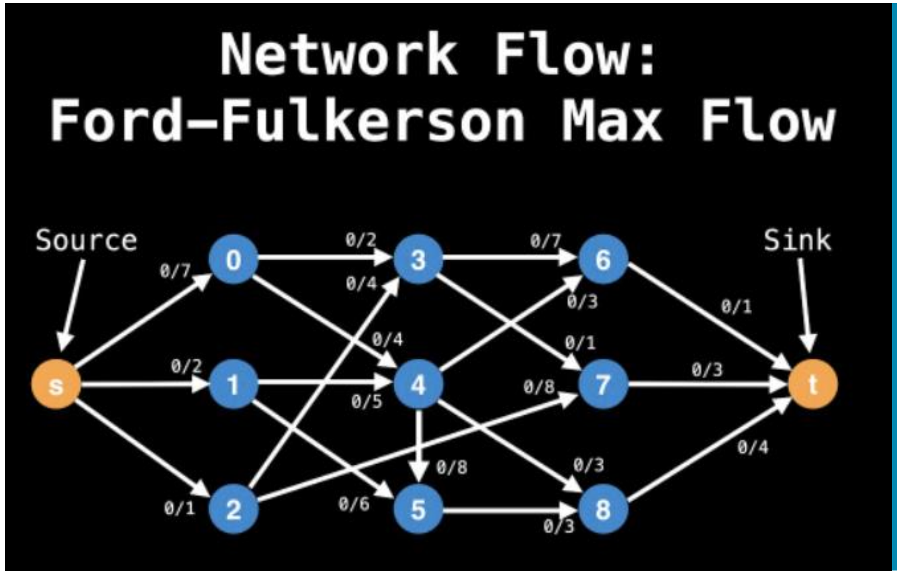


Complejidad Algorítmica

Unidad 2: Algoritmos voraces, programación dinámica y problemas P-NP

Módulo 11: Flujo Máximo en Redes

Complejidad Algorítmica

Semana 11

MÓDULO 11: Flujo Máximo en Redes

Contenido

1. Conceptos básicos – Flujos en redes
2. El problema del Flujo Máximo en redes
3. Algoritmo Ford – Fulkerson
4. Aplicaciones del Flujo en Redes

Preguntas

1. Conceptos básicos – Flujos en redes

En teoría de grafos:
¿Qué es **una red** y un **flujo de red**?

Red: En teoría de grafos, **una red** es un **grafo dirigido con pesos.**

Flujo de Red: es la cantidad de "algún contenido" que circula dentro de una red.

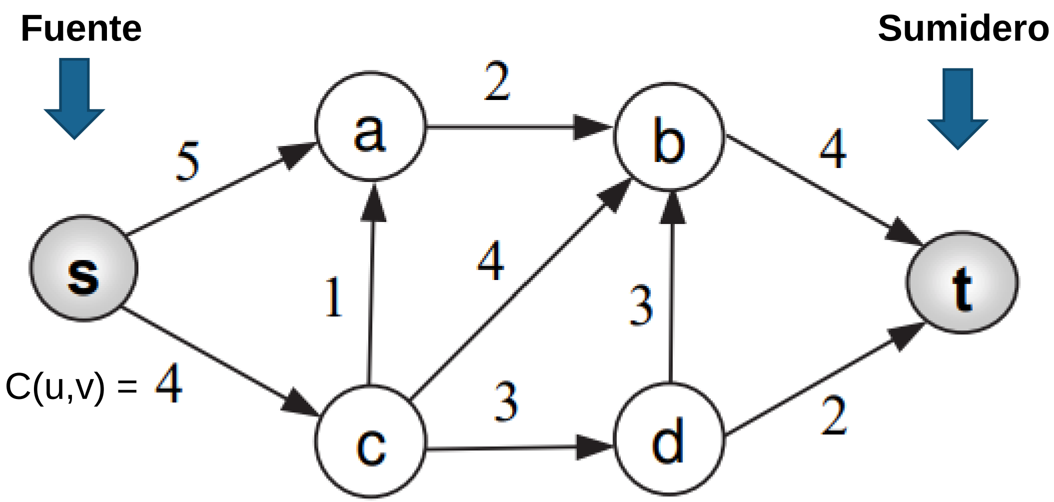
- Debemos entender que en una red, las aristas representan canales por los que puede circular cualquier elemento. Por ejemplo:
  - ❖ Datos
  - ❖ Agua
  - ❖ Autos
  - ❖ Corriente eléctrica, etc.
- Los pesos de las aristas representan la capacidad máxima de un canal:
  - ❖ Velocidad de una conexión
  - ❖ Volumen máximo de agua
  - ❖ Cantidad máxima de tráfico
  - ❖ Voltaje de una línea eléctrica, etc.
- Pero muchas veces es posible que la cantidad real de flujo de un canal sea menor a la considerada como valor en la arista.

1. Conceptos básicos – Flujos en redes

Características de flujo de una red

- Existe un **nodo inicial** llamado **Origen/Fuente (s)** y un **nodo final** llamado **Destino/Sumidero (t)**, ambos nodos pertenecen a **G**.
- Entre el origen y el destino existe una cantidad determinada de nodos interconectados entre sí a través de aristas.
- Estos nodos o vértices solo son uniones. **No consumen ni crean flujo**.
- Cada arista tiene una Capacidad definida no negativa **C(u,v)**.
- La capacidad máxima que puede transportar cada arista entre los nodos que conecta, puede variar de un arista a otra.
- Una arista solo podrá soportar un flujo menor o igual a su capacidad, así, si un flujo mayor quiere discurrir a través de una arista, solo una parte de dicho flujo (de valor igual a la capacidad) viajará a través de ella, y el resto deberá ir por otra arista que salga del mismo nodo. De no haber otra arista, entonces el flujo se verá reducido.

Grafo **G(V,A,W)**



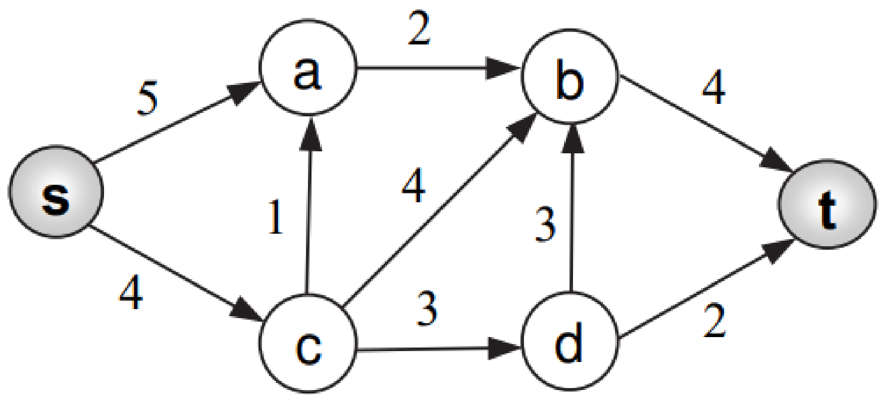
El flujo se preserva en todo momento.

“Todo flujo que entra debe salir”

1. Conceptos básicos – Flujos en redes

Representación del flujo de una red

- El siguiente grafo **G(V,A,W)** representa la circulación de algún elemento, que a través de cada una de sus aristas con una capacidad máxima de canal podrá transportar (fluir) algún contenido.



Tener en cuenta:

- Un nodo de origen **S** tiene todos los bordes salientes y ningún borde entrante.
- El nodo receptor o sumidero **T** tiene todos los bordes entrantes y ningún borde saliente.

Para cualquier red de flujo se debe cumplir:

- Para todos los nodos (excepto el fuente y sumidero), el flujo de entrada debe ser igual al flujo de salida.
- Existe una **función F** de flujo la cual indica la cantidad de flujo que transcurre por dicha arista (no podemos enviar más flujo a través de un borde que su capacidad).

$$\mathbf{f(u,v)} <= c(u,v)$$

- El nodo de origen es el único que produce flujo. Por lo tanto el flujo total de salida es:

$$\mathbf{|f|} = \text{Sum } f(s, v)$$

La salida total desde el vértice fuente debe ser igual a la entrada total al vértice receptor.

2. El problema del Flujo Máximo en redes

¿En que consiste el problema del flujo máximo?

El **problema del flujo máximo** consiste en **encontrar la máxima cantidad de flujo que puede ser llevado desde el nodo s al nodo t**.

Objetivo

Determinar la máxima cantidad de material u objetos (flujo) que puede fluir en la red desde la fuente (s) hacia el sumidero (t).

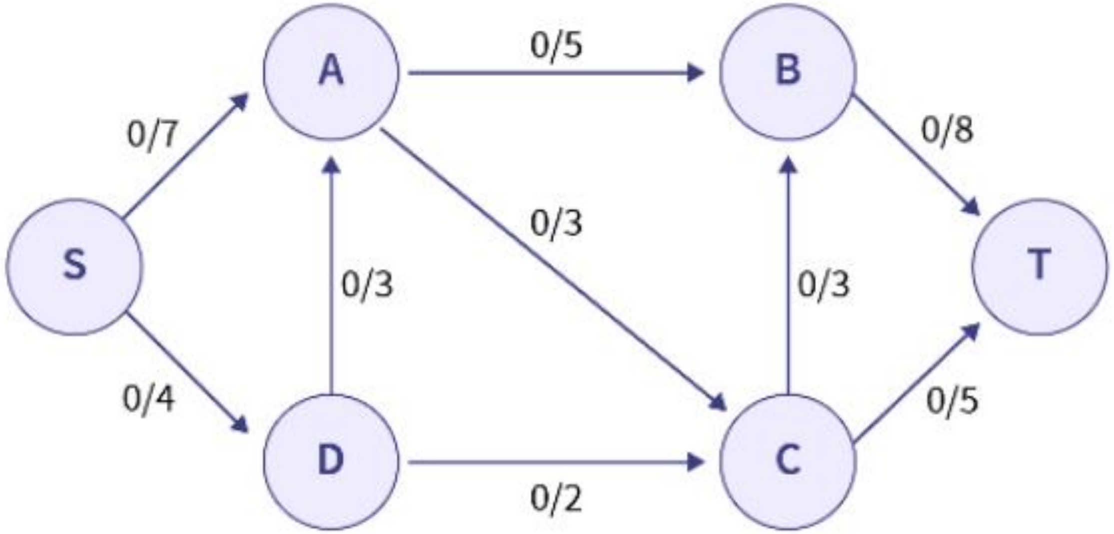
- Será muy importante conocer el valor del flujo máximo porque permite a la fuente saber exactamente cuánto producir y enviar a través de una ruta sin generar desperdicios.

Flujo Máximo

El flujo máximo es el valor máximo posible del flujo de la red donde el flujo se define como "suma de todo el flujo que se produce en la fuentes **s**, o suma de todo el caudal que se consume en el sumidero **t**.

2. El problema del Flujo Máximo en redes

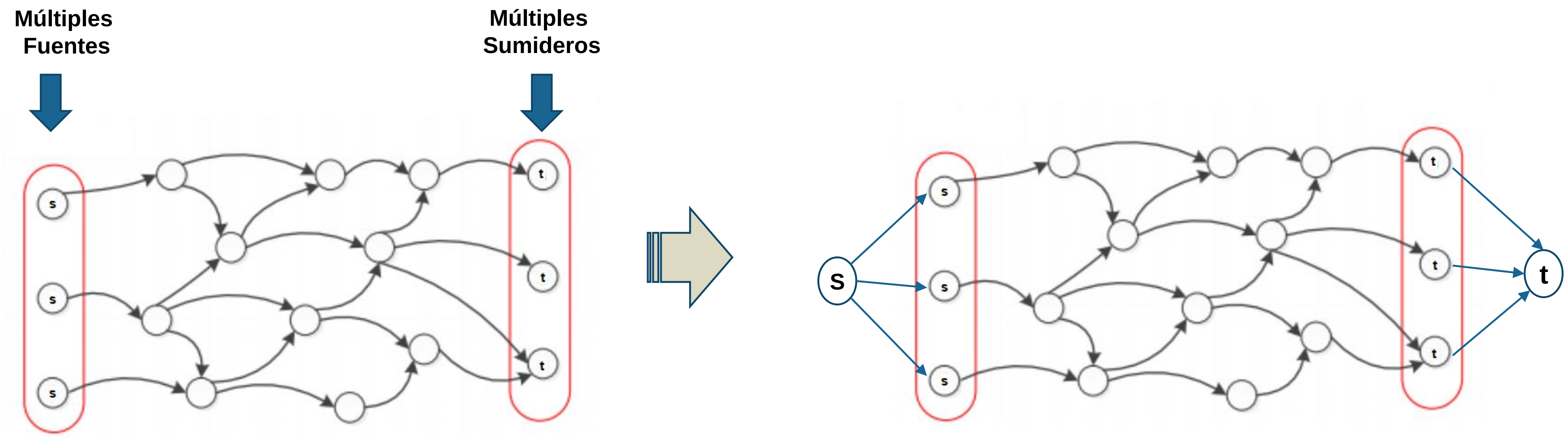
Representación de una red de flujo



- El primer valor de cada borde o arista representa la cantidad de flujo que lo atraviesa (que inicialmente se establece en 0).
- El segundo valor representa la capacidad real del borde o arista.

2. El problema del Flujo Máximo en redes

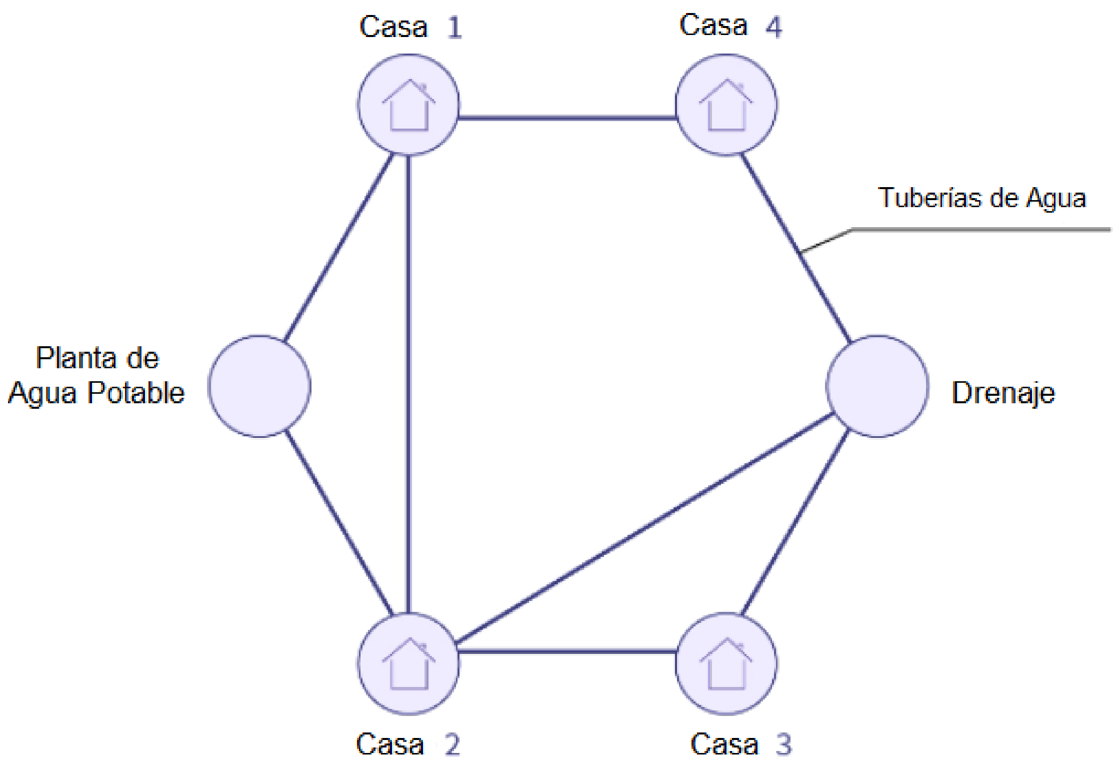
Representación de una red de flujo



- Cualquiera que sea la cantidad de orígenes o destinos, esta situación se puede reducir a un problema de flujo máximo ordinario (una sola fuente y un único sumidero).

2. El problema del Flujo Máximo en redes

Problema de la vida real de una Red de Flujo



- Podemos visualizar todos los bordes o aristas como tuberías de agua, donde su capacidad (aquí tubería) es la cantidad máxima de agua que puede fluir a través de la arista por unidad de tiempo.
- Aquí el caudal máximo es análogo a la planta de agua donde **S** es análoga a la planta de drenaje.

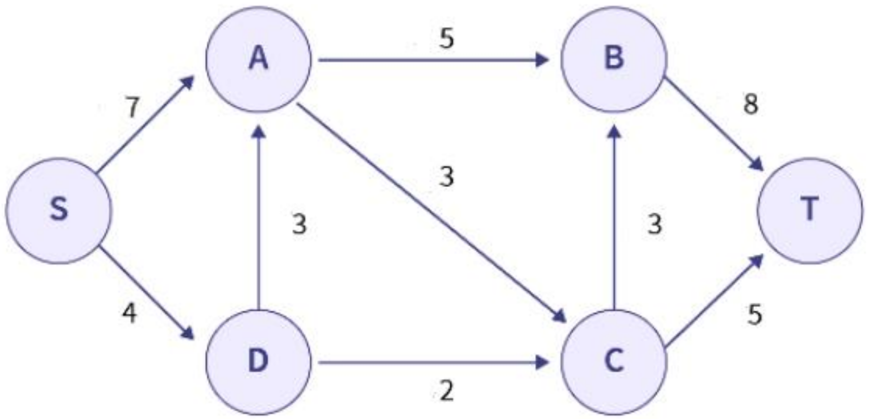
Cumple las condiciones:

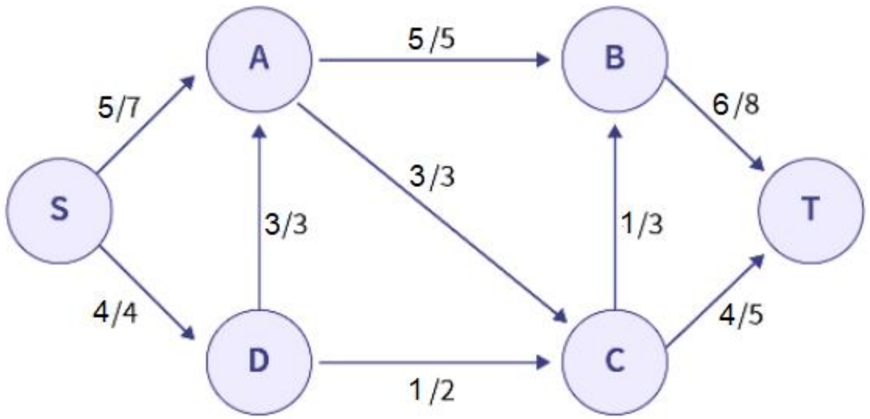
- Más agua que la capacidad de la tubería no puede fluir a través de ella.
- La entrada y salida de agua en cada nodo (casa) debe ser igual ya que el agua no puede desaparecer o aparecer mágicamente.

2. El problema del Flujo Máximo en redes

Ejemplo de representación del flujo máximo en una red

Grafo G(V,A)





Donde:

- G (V, A) es una red de transporte.
- Cada arista (u, v) tiene capacidad q y flujo f.
- Asumimos siempre que el flujo es compatible.

Flujo/Capacidad

f(u,v) <= c(u,v)

3. Algoritmo Ford – Fulkerson

¿Qué algoritmos solucionan el problema del Flujo Máximo?

- El principal método de solución del problema del flujo máximo es el **Algoritmo de Ford-Fulkerson.**

En 1956, con el problema de flujo máximo Ford-Fulkerson establecieron el famoso teorema del flujo máximo - mínimo corte.

Lester Randolph Ford Jr.

Delbert Ray Fulkerson

Es un algoritmo codicioso (greedy) que implica:

- Encontrar una ruta desde el nodo de origen (s) al nodo sumidero (t)
- Y si el camino existe entonces aumentamos el flujo tanto como sea posible para que a lo largo de cualquier borde o arista no exceda su capacidad.

3. Algoritmo Ford – Fulkerson

Veamos el algoritmo

E = número de aristas en la red de flujo.

f (e)= flujo a lo largo del borde o arista.

c (e)= Capacidad del borde o arista.

- Inicializar maxFlow con 0, es decir, **maxFlow=0**
- Usar un algoritmo **BFS/DFS** para buscar un camino **P** entre los nodos **s** y **t**. La existencia de **P**, significaría que por cada arista **e**, se cumple que f(e) < c (e). Si el camino **P** existe entonces:
  - **flowCanBeAdded** = min(c(e)−f(e)) para cada arista **e** en **P**.
  - **maxFlow** = maxFlow + flowCanBeAdded
  - Por cada arista **e** en **P**, f(e)=f(e)+flowCanBeAdded
- En el otro caso (si no existe ningún camino entre s y t) devolver el maxFlow.

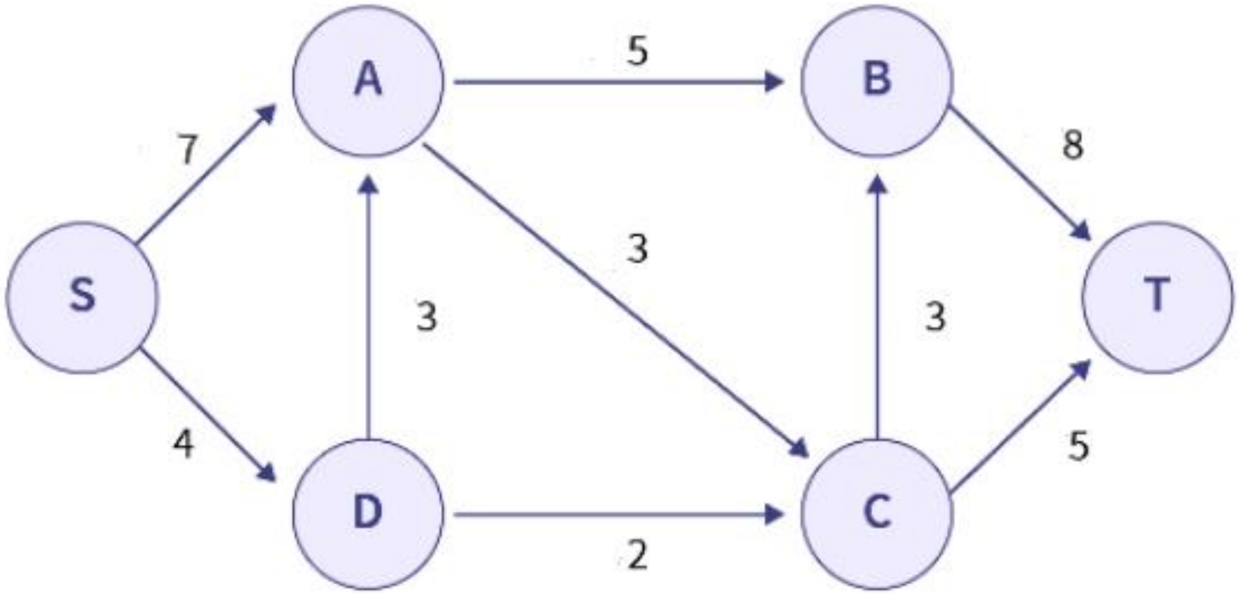
Pseudocodigo

```
FordFulkerson(Graph G, Node s, Node t):
   Initialize flow of all edges e to 0.
   while(there is augmenting path(P) from s to t
   in the residual graph):
      Find augmenting path between s and t.
      Update the residual graph.
      Increase the flow.
   return
```

Veamos el algoritmo en un ejemplo

3. Algoritmo Ford – Fulkerson

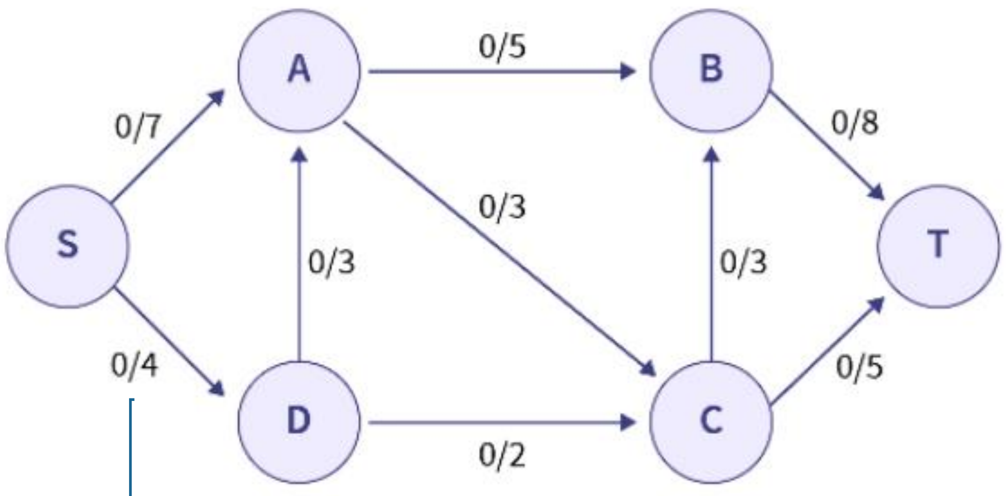
EJEMPLO #1: Encuentre el máximo flujo de **s** a **t** en el siguiente gráfico.



3. Algoritmo Ford – Fulkerson

EJEMPLO #1: Encuentre el máximo flujo de **s** a **t** en el siguiente gráfico.

Grafo residual G’



c(e)−f(e) = Capacidad residual = lo que falta por fluir

- Podemos ver que el **flujo inicial** de todos los caminos es 0.
- Ahora buscaremos una ruta (camino) de aumento en la red.

Camino de aumento

Es una trayectoria desde el nodo fuente **s** al nodo sumidero **t** que puede conducir más flujo.

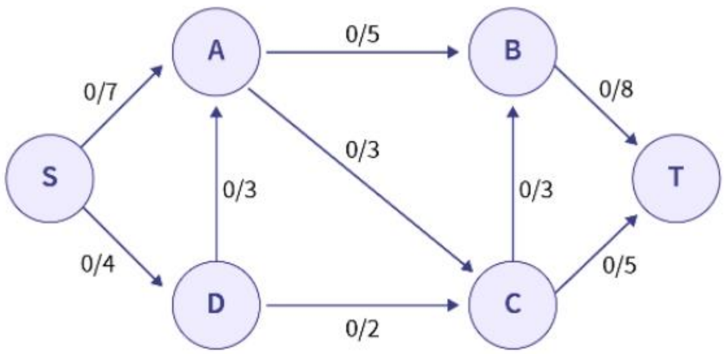
Capacidad residual

Es la capacidad restante después de llevar el flujo

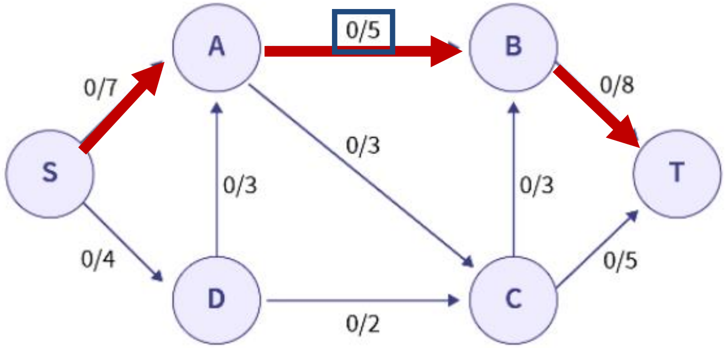
3. Algoritmo Ford – Fulkerson

Uno de esos caminos es el que presenta capacidades residuales como 7, 5, y 8. Su mínimo capacidad residual es 5, por lo que según el método de **Ford-Fulkerson** aumentaremos un flujo de 5 a lo largo de todas las aristas del camino.

Grafo residual G'



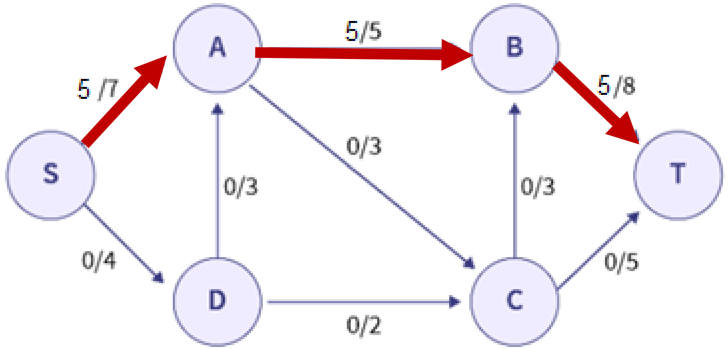
1. Se comienza con f(u,v) =0 para cada par de nodos.



2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

Hallar un C(p)

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e))**

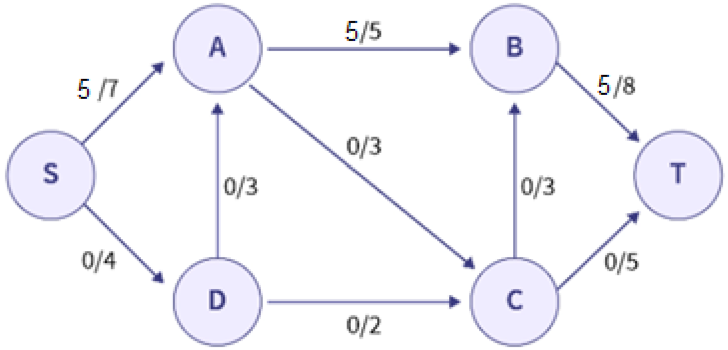


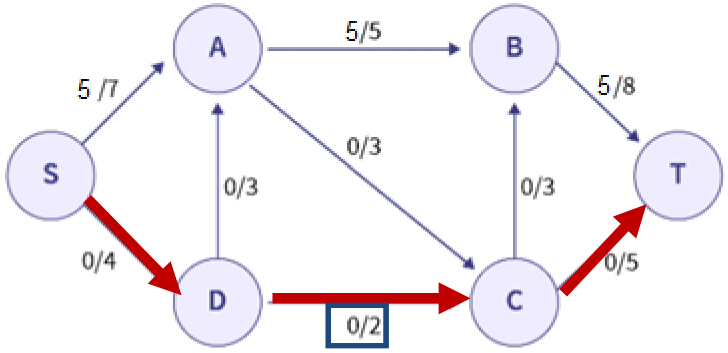
C(p) = 5 acumulamos este flujo en cada arista del camino.

3. Algoritmo Ford – Fulkerson

- Ahora, buscaremos otras posibles rutas (caminos) de aumento, una de esas rutas puede ser: **s→D→C→t** con capacidades residuales como 4, 2, y 5 de los cuales 2 es el mínimo.
- Entonces aumentaremos un flujo de 2 a lo largo del camino.

Grafo residual G'

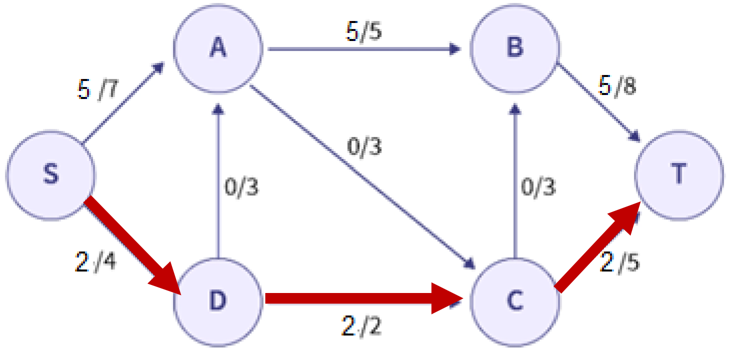




2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

<u>Hallar un C(p)</u>

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e))**

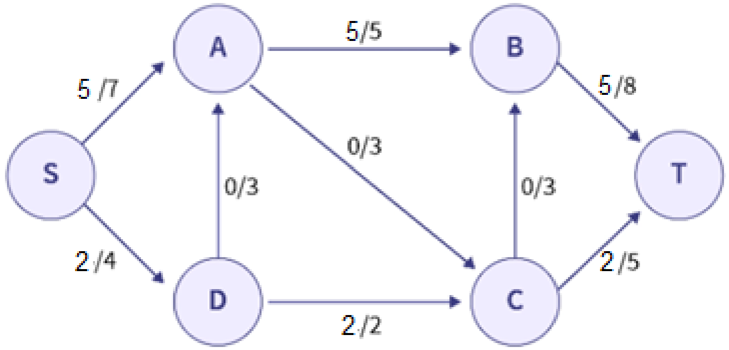


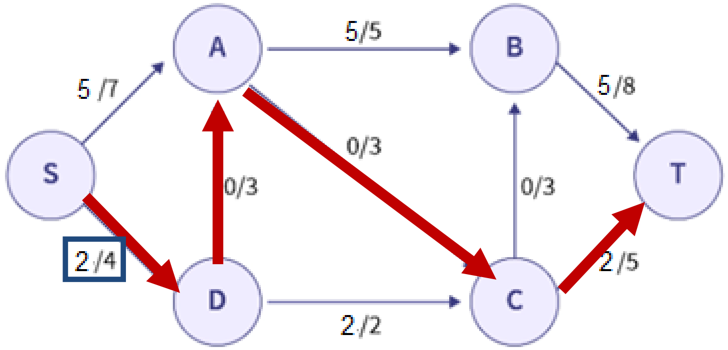
C(p) = 2 acumulamos este flujo en cada arista del camino.

3. Algoritmo Ford – Fulkerson

- Uno de esos caminos puede ser **s→D→A→C→t** con capacidades 2,3,3, y 3 de los cuales 2 es el mínimo por lo que aumentará el flujo en 2 a lo largo del camino.

Grafo residual G'

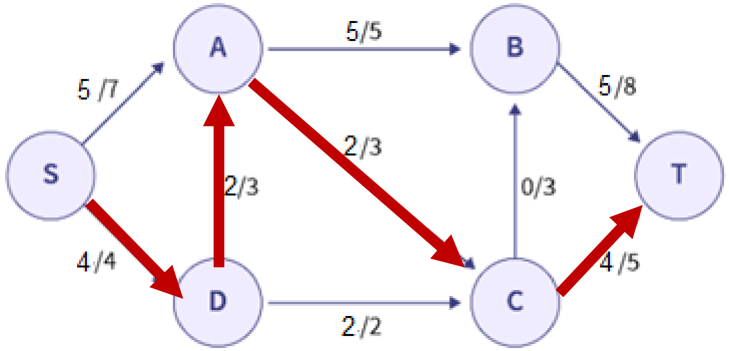




2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

<u>Hallar un C(p)</u>

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e))**

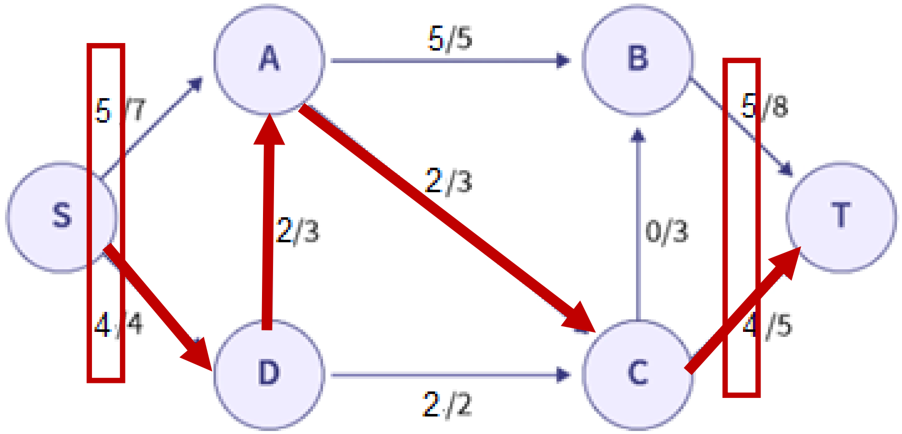


C(p) = 2 acumulamos este flujo en cada arista del camino.

3. Algoritmo Ford – Fulkerson

- En este ultimo grafo residual, podemos observar que hemos alcanzado un flujo máximo en las aristas que unen los nodos s->t.

Grafo residual G'

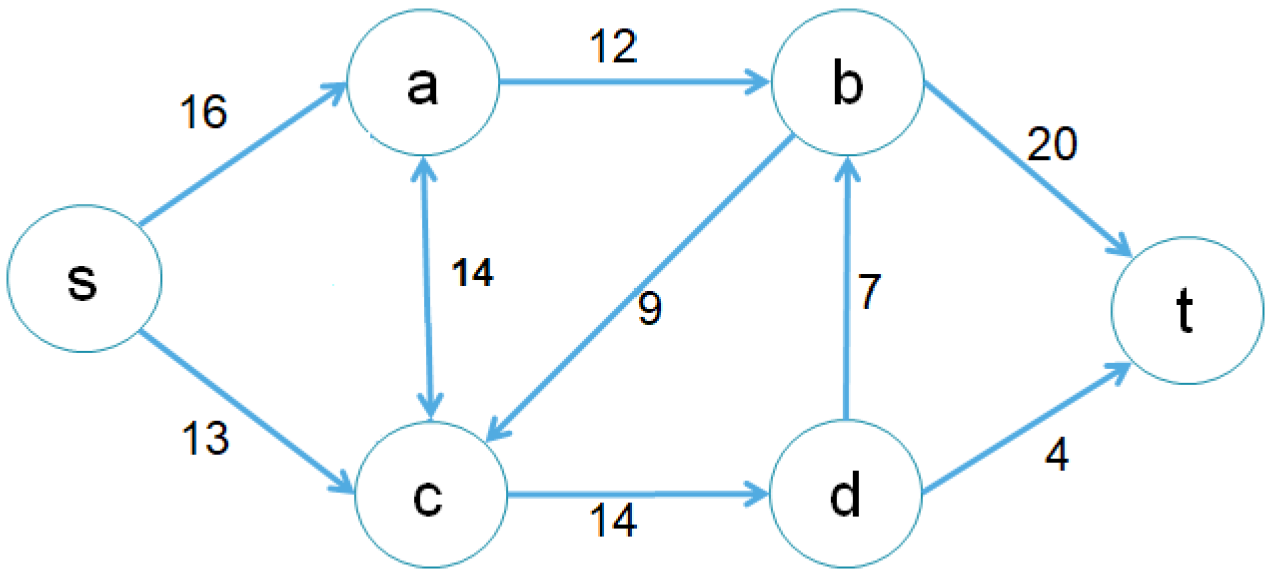


Ahora no hay más rutas de aumento posibles en la red, por lo que el flujo máximo es el flujo total que sale de la fuente o el flujo total que entra en el sumidero, es decir

Flujo máximo = 5 + 4 = 9 = Flujo total 5+4 (que sale de la fuente **s**) = Flujo total 5 + 4 (que entra al sumidero **t**)

3. Algoritmo Ford – Fulkerson

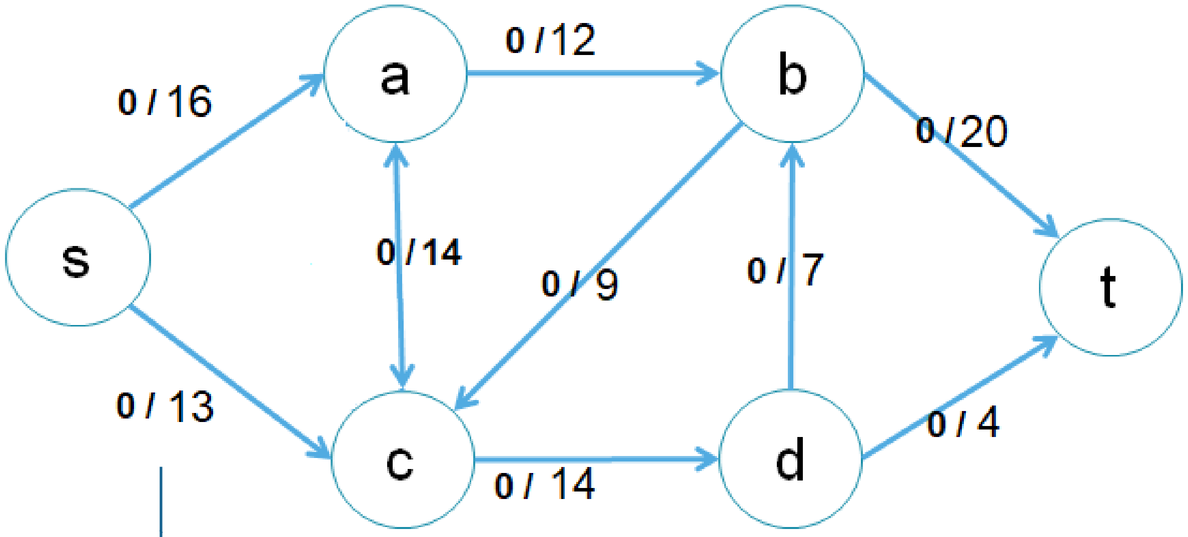
EJEMPLO #2: Encuentre el máximo flujo de **s** a **t** en el siguiente gráfico.



3. Algoritmo Ford – Fulkerson

EJEMPLO #2: Encuentre el máximo flujo de **s** a **t** en el siguiente gráfico.

Grafo residual G'



c(e)−f(e) = Capacidad residual = lo que falta por fluir

- Podemos ver que el **flujo inicial** de todos los caminos es 0.
- Ahora buscaremos una ruta (camino) de aumento en la red.

Camino de aumento

Es una trayectoria desde el nodo fuente **s** al nodo sumidero **t** que puede conducir más flujo.

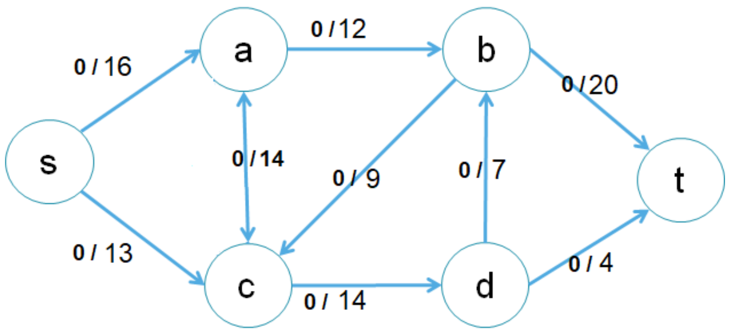
Capacidad residual

Es la capacidad restante después de llevar el flujo

3. Algoritmo Ford – Fulkerson

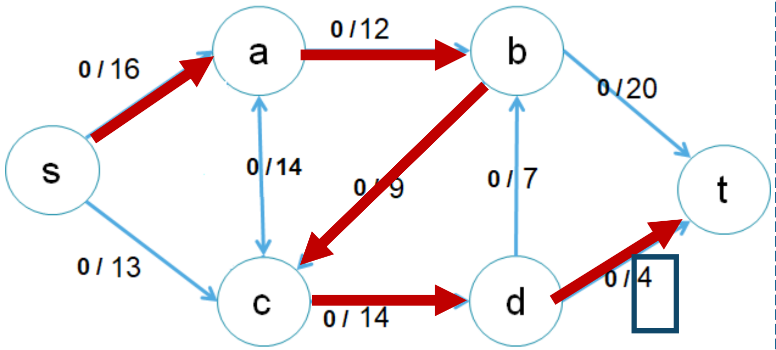
EJEMPLO #2: Encuentre el máximo flujo de **s** a **t** en el siguiente gráfico.

Grafo residual G'



1. Se comienza con f(u,v) =0 para cada par de nodos.

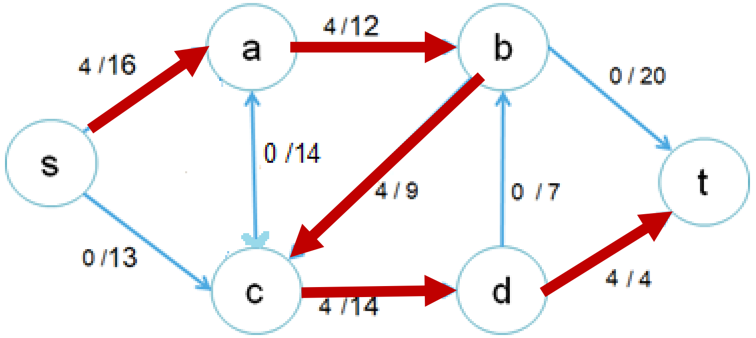
- Encontrar un camino de **s** a **t**



2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

<u>Hallar un C(p)</u>

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e))**

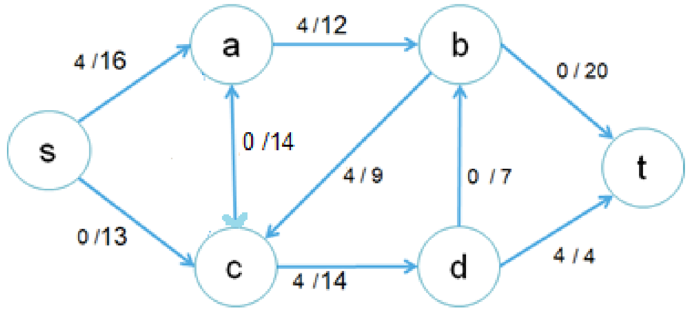


C(p) = 4 acumulamos este flujo en cada arista del camino.

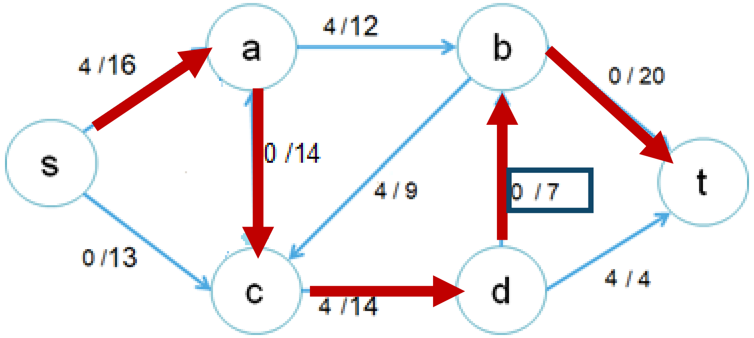
3. Algoritmo Ford – Fulkerson

3. Se repite el proceso previo hasta no encontrar un camino de aumento.

Grafo residual G'



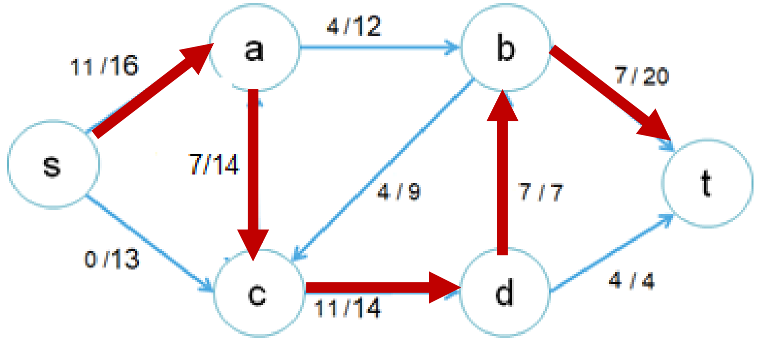
- Encontrar un camino de **s** a **t**



2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

Hallar C(p)

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e)) = 7 – 0 = 7**

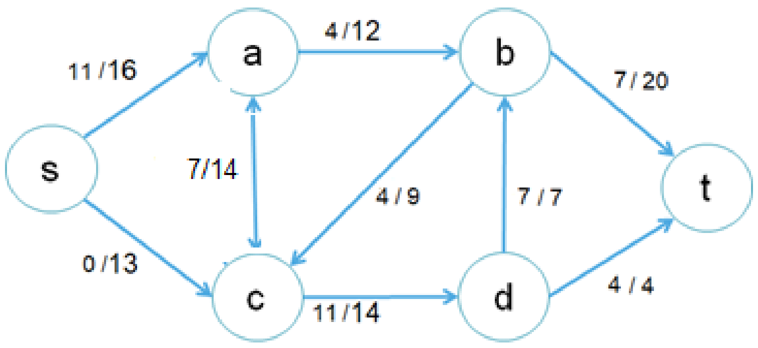


C(p) = 7 acumulamos este flujo en cada arista del camino.

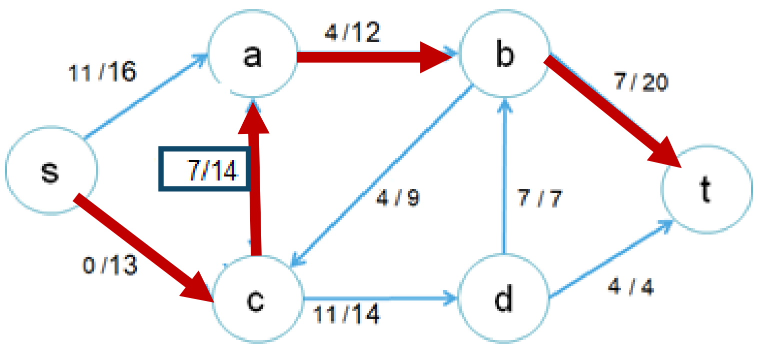
3. Algoritmo Ford – Fulkerson

3. Se repite el proceso previo hasta no encontrar un camino de aumento.

Grafo residual G'



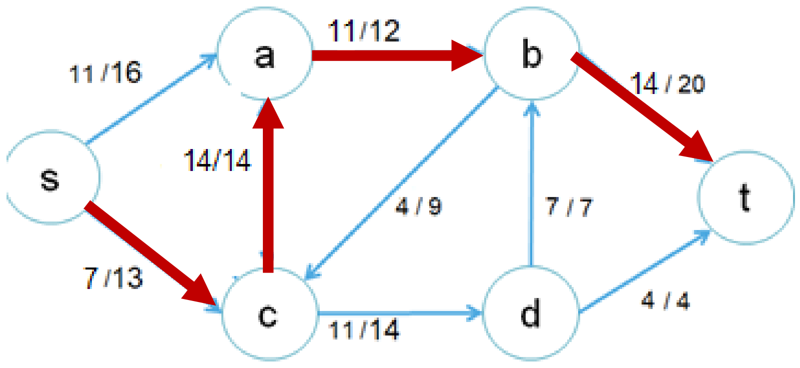
- Encontrar un camino de **s** a **t**



2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

Hallar C(p)

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e)) = 14 – 7 = 7**

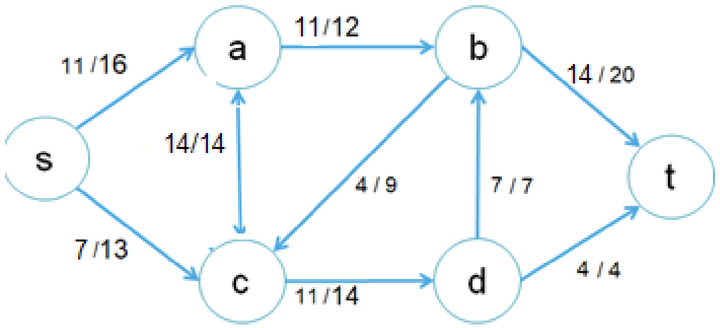


C(p) = 7 acumulamos este flujo en cada arista del camino.

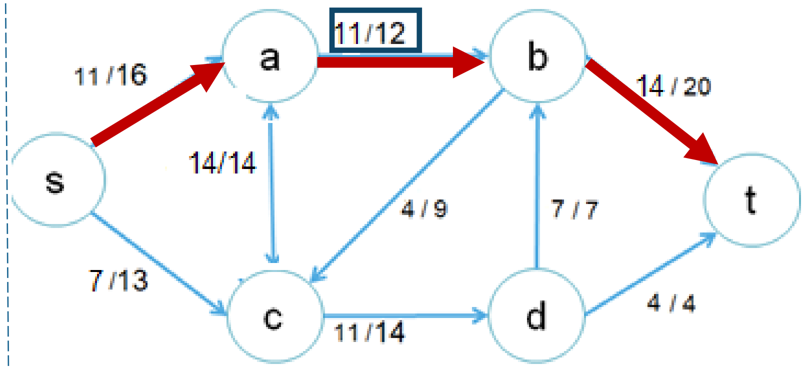
3. Algoritmo Ford – Fulkerson

3. Se repite el proceso previo hasta no encontrar un camino de aumento.

Grafo residual G'



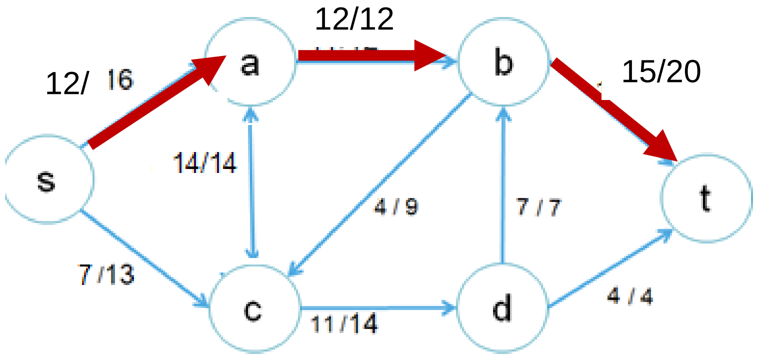
- Encontrar un camino de **s** a **t**



2. En cada iteración se incrementa el valor del flujo buscando un **camino de aumento** (seleccionando el camino que lleve más flujo).

Hallar C(p)

donde **C(p)** sea el mínimo valor en las aristas del camino **min(c(e)−f(e)) = 12 – 11 = 1**



S-> a = 12/16

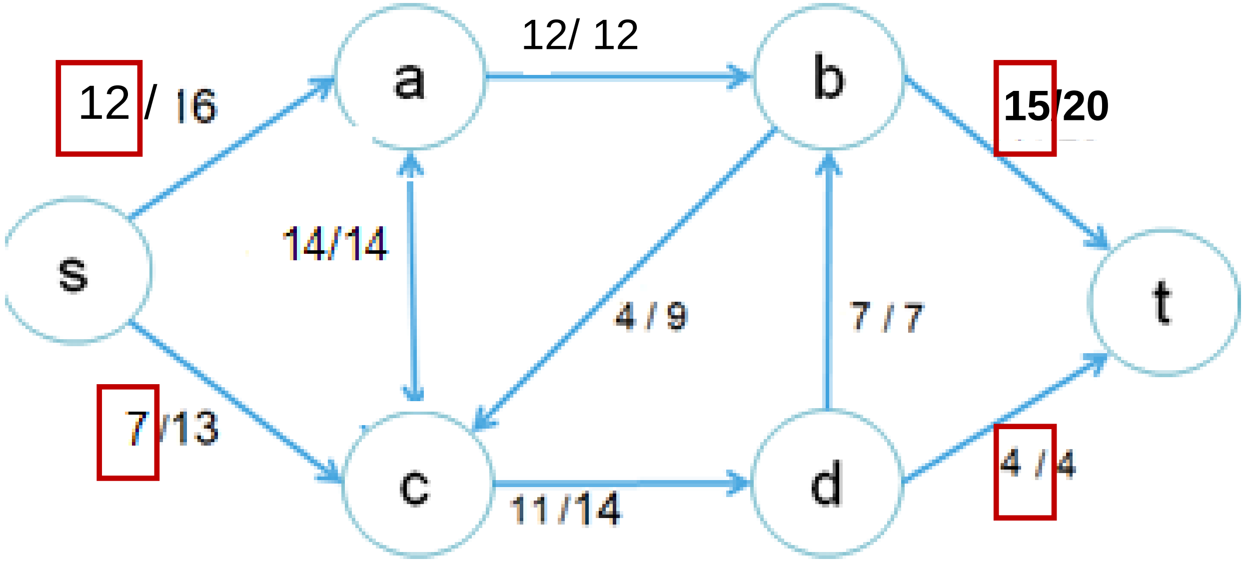
b-> t = 15/20

a-> b = 12/12

C(p) = 1 acumulamos este flujo en cada arista del camino.

3. Algoritmo Ford – Fulkerson

Grafo residual G’



Flujo máximo = 15 + 4 = 19

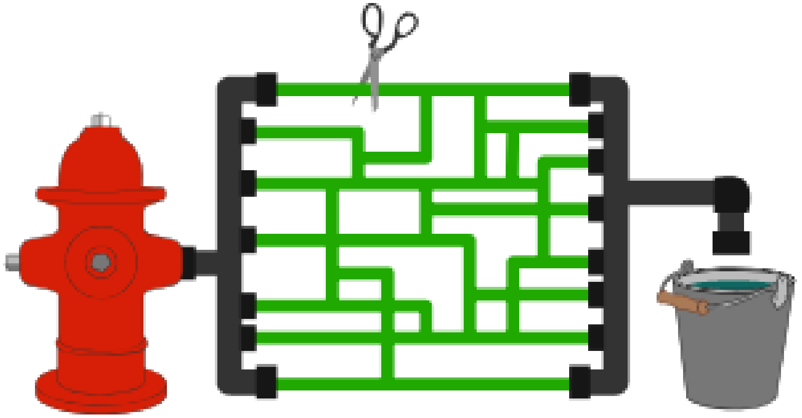
Ahora no hay más rutas de aumento posibles en la red, por lo que el flujo máximo es el flujo total que sale de la fuente o el flujo total que entra en el sumidero, es decir

Flujo máximo = 15 + 4 = 19 = Flujo total 12+7 (que sale de la fuente **s**) = Flujo total 15 + 4 (que entra al sumidero **t**)

3. Aplicaciones del Flujo Máximo en Redes

Principales aplicaciones

- En movimientos de tráfico, para encontrar cuánto tráfico puede moverse de una ciudad A a la ciudad B.
- En los sistemas eléctricos, para encontrar la corriente máxima que podría circular por los cables sin provocar ningún cortocircuito.
- En el sistema de gestión de agua/alcantarillado, para encontrar la capacidad máxima de la red.



PREGUNTAS

Dudas y opiniones