

# El algoritmo de Dinitz o Dinic (y Even)

Daniel Penazzi

28 de abril de 2021

# Tabla de Contenidos

## Introducción

Un poco de historia

## Networks auxiliares

Idea básica

Network por niveles

## Ejemplo

Primer network auxiliar

Segundo network auxiliar

Tercer Network Auxiliar

Cuarto Network Auxiliar

## Observaciones

## Edmonds y Karp

- Vimos que Edmonds-Karp funciona y termina en tiempo polinomial  $O(nm^2)$ .
- Para networks ralos ( $n \simeq m$ ), esto da una complejidad de  $O(n^3)$ .
- Esa es la complejidad de los algoritmos mas rápidos conocidos.
- Pero para networks densos ( $m \simeq n(n-1)/2$ ), da una complejidad  $O(n^5)$ , no tan buena.
- Veremos otros algoritmos que permiten reducir esa complejidad..

## Edmonds y Karp

- El problema con E-K es que usa BFS (y, como vimos, esto es clave para que termine)
- Con lo cual se puede producir mucho cálculo que no se usa.
- BFS busca todos los caminos posibles en paralelo, hasta llegar a  $t$  y de todos esos caminos, sólo nos quedamos con uno.
- En la siguiente iteración, es muy posible que repitamos muchos de los mismos cálculos que ya hicimos.
- ¿Podríamos usar una estructura para ir “guardando” estos cálculos y no tener que repetirlos?

# Dinitz

- Esta idea se le ocurrió a un ruso, Yefim Dinitz.
- Dinitz, **independientemente** de Edmonds y Karp, “mejoró” Edmonds-Karp.
- Es decir, mejoró un algoritmo que no conocía.
- De hecho, el algoritmo de Dinitz fue publicado ANTES que el de Edmonds-Karp.
- Pero obviamente en el occidente el de Edmonds-Karp se conoció antes.

## Dinitz

- De hecho, la historia es mas curiosa: Dinitz no sólo no conocia el algoritmo de Edmonds-Karp: TAMPOCO conocia (al principio) a Ford-Fulkerson!
- Mas aún: Dinitz inventó su algoritmo para resolver un EJERCICIO DE UNA CLASE de la cual era alumno. !
- Pero como no habian visto Ford-Fulkerson, el creia que con el algoritmo Greedy bastaba.
- Lo que hizo fue mejorar la complejidad de Greedy.
- Luego, al enterarse como era Ford-Fulkerson en la clase, aplicó su idea con las ideas de Ford-Fulkerson.
- Años despues, Dinitz dijo que si hubiera conocido Ford-Fulkerson de entrada, la idea de su algoritmo quizás no se le hubiera ocurrido.

## Dinitz o Dinics o Dinic

- La idea clave de ir guardando trabajo hecho en una estructura especial para ser usada en futuras iteraciones resultó ser muy fructifera y luego surgieron numerosos otros algoritmos, primero en Rusia y luego en otros lados, usando esa idea de Dinitz.
- Dinitz publicó su algoritmo (en ruso, obvio) en un paper sumamente denso de sólo 4 paginas. pues ese era el límite del journal donde lo publicó.
- Luego algunos refinamientos fueron hechos por otro ruso, Karzanov, también publicados en un paper denso de 4 páginas.
- En occidente casi nadie les prestó atención y los pocos que lo hicieron no podían entender esos papers.

## Dinitz o Dinic

- Esos papers fueron leídos por un israelí, Shimon Even, quien tardó un buen tiempo en entenderlos pero perseveró y luego de entenderlos popularizó el concepto en el occidente.
- Al parecer Even, al cambiar del alfabeto cirílico al latino, lo escribió como “Dinic” en vez de el correcto “Dinitz”, así que es más conocido de esa forma en occidente.
- Mas aún, Even no entendió del todo una parte del algoritmo de Dinitz, pero mientras trataba de entenderla, se dio cuenta que podía simplificar el algoritmo para no tener que usar esa parte.
- Y como el lo popularizó en el occidente, la versión que se conoce en el occidente es esa.







## Network auxiliar

- Ahora bien, por mas que tengamos guardados “todos” los caminos de longitud mínima, no podemos usarlos a todos para aumentar el flujo, porque algunos de esos compartiran lados, y si usamos un camino A pej, es posible que luego no podamos usar un camino B.
- Entonces el problema queda, luego de tener toda la información guardada sobre todos los posibles caminos aumentantes, ¿cómo seleccionar de entre ellos los que efectivamente vamos a usar?
- Esa estructura auxiliar puede tomar varias formas, pero en la forma original de Dinitz, y la mas fácil de explicar, esos datos se guardan en un network, que llamaremos “network auxiliar”.

## Network auxiliar

- En breve daremos la definición precisa de network auxiliar y como construirlo, pero por ahora basta con asumir que hay un network auxiliar que guarda mucha información.
- La idea clave y la belleza de Dinitz es que una vez construido el network auxiliar usando BFS, encontramos los caminos aumentantes que vamos a usar.... corriendo Greedy en ese network auxiliar usando DFS !
- (usar BFS dentro del network auxiliar seria una idiotez, porque justamente queremos evitar el desperdicio de trabajo de usar BFS).
- Se usa el network auxiliar hasta que no se puede usar mas, y luego se contruye otro a partir del nuevo flujo.

## Esquema básico de Dinitz

■ La estrategia de Dinitz entonces es:

- 1 Construir un network auxiliar (usando BFS).
- 2 Correr Greedy con DFS en el network auxiliar hasta no poder seguir.
- 3 Usar el flujo obtenido en el network auxiliar para modificar el flujo en el network original.
- 4 Repetir [1] con el nuevo flujo, hasta que, al querer construir un network auxiliar, no llegamos a  $t$ .

## Un par de detalles

- En realidad, ese esquema es la forma en que uno lo plantea teóricamente para poder hacer análisis.
- En la práctica, cuando encontramos un camino dirigido en el network auxiliar, como ese camino estará representando un camino aumentante de Ford-Fulkerson en el network original, se puede directamente cambiar el flujo original en el network original.
- En el network auxiliar, como se usa Greedy, nunca se des-satura un lado. en el network auxiliar
- Obviamente como algunos de los lados del network auxiliar van a estar representando lados backwards en el network original, al mandar flujo POR el lado en el network auxiliar, estaríamos DEVOLVIENDO flujo en el original.
- Asi que los lados siguen pudiendo des-saturarse, es sólo en el network auxiliar que no se des-saturan.

- ¿Pero qué pasa si sólo hay un camino de longitud mínima?
- En ese caso habremos malgastado esfuerzo, pero en general hay múltiples caminos de longitud mínima.
- Si siempre tenemos un sólo camino de longitud mínima entonces es fácil ver que Edmonds-Karp termina en  $O(nm)$  en vez de  $O(nm^2)$ .
- La complejidad de Dinic en este caso de que en cada etapa sólo hay un camino de longitud mínima también es  $O(nm)$ , el esfuerzo “extra” de crear el network auxiliar queda oculto en la notación “ $O$ ”.

## Flujos bloqueantes

- Luego de la idea original de Dinitz, algunas partes fueron refinadas, en particular Alexander Karzanov introdujo el concepto de “flujo bloqueante” que facilita algunas presentaciones y análisis del algoritmo.

### Definición:

Llamaremos a un flujo en un network **bloqueante** o **saturante** si todo camino DIRIGIDO desde  $s$  a  $t$  tiene al menos un lado saturado. (es decir con  $c(\overrightarrow{xy}) = f(\overrightarrow{xy})$ ).

- En otras palabras, si cuando queremos usar Greedy en el network, no llegamos a  $t$ .



## Algoritmos tipo Dinic

- Asi, podemos resumir la idea básica de Dinitz reformulada por Karzanov como:
  - 1 Construir un network auxiliar (usando BFS).
  - 2 Encontrar un flujo bloqueante en el network auxiliar.
  - 3 Usar ese flujo bloqueante del network auxiliar para modificar el flujo en el network original.
  - 4 Repetir [1] con el nuevo flujo, hasta que, al querer construir un network auxiliar, no llegamos a  $t$ .

## Algoritmos tipo Dinic

- Luego de esta idea de Dinitz, surgieron muchos otros algoritmos, “tipo”Dinic.
- También usan un network auxiliar pero cambian la forma de hacer [2], es decir, cambian la forma de encontrar un flujo bloqueante, usando un método distinto del Greedy original de Dinitz.
- De hecho hay algoritmos que encuentran un flujo bloqueante sin necesidad de ir encontrando caminos dirigidos desde  $s$  a  $t$ .
- Veremos uno de esos mas adelante.

## Actualización del network auxiliar

- Si bien teóricamente estamos diciendo que vamos a buscar un flujo bloqueante en el network auxiliar, ese es el refinamiento de Karzanov.
- Dinitz originalmente hacia otra cosa, que es lo que haremos en la práctica porque es mas fácil de implementar.
- Observemos que, luego de cada camino, queda al menos un lado saturado en ese camino, el cual no es mas utilizable en el network auxiliar.
- Dinitz toma ventaja de esto **cambiando** el network auxiliar luego de cada camino, borrando esos lados.
- Asi, en realidad el algoritmo se detiene cuando se borraron tantos lados que  $s$  quedó desconectado de  $t$ . (lo cual es equivalente a tener un flujo bloqueante en el network auxiliar inicial).

## Diferencia entre la version rusa y la occidental de Dinitz

- Luego de borrar los lados saturados en un camino, se puede actualizar aun mas el network auxiliar, buscando lados o vértices que ya no se pueden usar debido a haber borrado esos lados.
- Dinitz original hace esto inmediatamente luego de cada camino, mientras que la versión de Evens es mas “lazy” y sólo los borra mientras está haciendo la busqueda DFS y llega a uno de ellos.
- Mas adelante discutiremos esto mas en detalle.



## Network auxiliar, vértices

- Cómo se define el network auxiliar, usando el network original y un flujo  $f$ :
- Vértices: el conjunto de vértices es  $V = \cup_{i=0}^r V_i$ , donde los  $V_i$  son:
  - Sea  $r = d_f(s, t)$  donde  $d_f$  es la función definida en la prueba de Edmonds-Karp.
    - Es decir,  $r$  es la distancia entre  $s$  y  $t$  usando caminos aumentantes.
  - Para  $i = 0, 1, \dots, r - 1$ , definimos  $V_i = \{x : d_f(s, x) = i\}$ .
    - Observar que entonces  $V_0 = \{s\}$ .
  - Definimos  $V_r = \{t\}$

## Network auxiliar

- Observar que  $\{x : d_f(s, x) = r\}$  puede tener mas vértices que sólo  $t$ , pero  $V_r$  se define como  $\{t\}$ .
- **Lados y capacidades:**  $\overrightarrow{xy}$  es un lado del network auxiliar si:
  - $x \in V_i, y \in V_{i+1}$
  - $y$ :
    - 1  $\overrightarrow{xy}$  es un lado del network original con  $f(\overrightarrow{xy}) < c(\overrightarrow{xy})$ .  
o:  
2  $\overrightarrow{yx}$  es un lado del network original con  $f(\overrightarrow{yx}) > 0$ .
- En el caso de [1], la capacidad de  $\overrightarrow{xy}$  en el network auxiliar será  $c(\overrightarrow{xy}) - f(\overrightarrow{xy})$ , y en el caso de [2], la capacidad del lado  $\overrightarrow{xy}$  en el network auxiliar será  $f(\overrightarrow{yx})$ .





## Construcción del Network auxiliar

- Es decir, la forma de construirla es tomar como  $V_0$  a  $\{s\}$ .
- Y luego ir construyendo una cola a partir de  $s$  al estilo Edmonds-Karp.
- Y si  $x$  agrega a  $z$  y  $x$  está en  $V_i$ , entonces  $z$  está en  $V_{i+1}$ .
- Excepto que en Edmonds-Karp si un vértice  $x$  va a agregar a un vértice  $z$  a la cola, pero  $z$  ya estaba en la cola, no se hace nada.
- En cambio aca, si  $z$  ya está agregado, si bien  $z$  no se vuelve a agregar, el lado  $\overrightarrow{xz}$  si se agrega al network auxiliar, siempre y cuando la distancia de  $z$  a  $s$  sea uno mas que la distancia de  $x$  a  $s$ .

## Construcción del Network auxiliar

- Si en algún momento llegamos a  $t$ , no paramos inmediatamente, pues podría haber mas lados que lleguen a  $t$ .
- Pero borramos todos los vértices que ya hubieramos incluido en el mismo  $V_r$  en el cual estamos poniendo a  $t$  (y los lados que terminaban en ellos).
- Y de ahí en mas no agregamos mas vértices, sólo lados entre vértices de  $V_{r-1}$  y  $t$ .

## Ejemplo

- Veamos un ejemplo.
- Antes de darlo, hay que aclarar que en realidad no lo vamos a dar **exactamente** como se implementaría el algoritmo en la versión original de Dinitz, o en la versión “occidental” de Ever.
- Luego explicaremos esos detalles, pero para dar un ejemplo pequeño como el que veremos, no hacen falta meterse con ciertos detalles de implementación, los cuales discutiremos luego.



## NA1

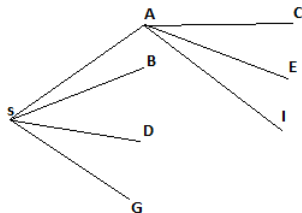
$\overrightarrow{sA} : 8, \overrightarrow{sB} : 7, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 9, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4 \dots$

- Como es el primer network auxiliar y el flujo inicial es 0, las capacidades de todos los lados del network auxiliar coinciden con las capacidades en el network original, así que no las escribo.
- Network auxiliar empieza con  $V_0 = \{s\}$ .
- $s$  agrega a sus vecinos  $A, B, D$  y  $G$ , que forman  $V_1$  y se agregan al network auxiliar los lados  $\overrightarrow{sA}, \overrightarrow{sB}, \overrightarrow{sD}, \overrightarrow{sG}$ .

## NA1

$\overrightarrow{sA} : 8, \overrightarrow{sB} : 7, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 9, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4$   
 $\overrightarrow{BH} : 9, \overrightarrow{BJ} : 9, \dots$

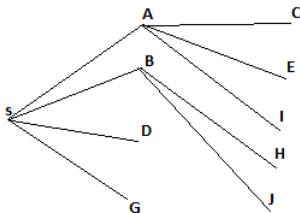
- A agrega a  $C, E$  e  $I$ , creando el conjunto  $V_2$  con ellos, y se agregan los lados  $\overrightarrow{AC}, \overrightarrow{AE}, \overrightarrow{AI}$ .



- Sigue en la cola  $B$ , con vecinos  $H, J$

## NA1

- $A$  agrega a  $C$ ,  $E$  e  $I$ , creando el conjunto  $V_2$  con ellos, y se agregan los lados  $\overrightarrow{AC}$ ,  $\overrightarrow{AE}$ ,  $\overrightarrow{AI}$ .

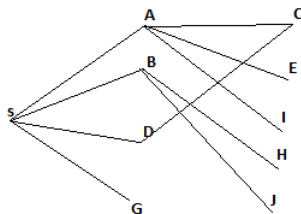


- Sigue en la cola  $B$ , con vecinos  $H$ ,  $J$  los cuales son agregados a  $V_2$  y los lados  $\overrightarrow{BH}$ ,  $\overrightarrow{BJ}$  al conjunto de lados.

## NA1

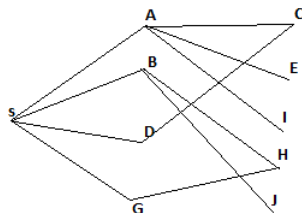
- Sigue  $D$ , que tiene vecinos  $A$ ,  $C$  y  $G$ . Los tres ya están, pero:

- $D \in V_1$ ,  $C \in V_2$ , así que es “legal” agregar el lado  $\overrightarrow{DC}$  al network auxiliar.
- $D, A, G \in V_1$ , entonces NO agregamos los lados  $\overrightarrow{DA}$ ,  $\overrightarrow{DG}$ , pues sólo agregamos lados de un nivel  $V_i$  a un nivel  $V_{i+1}$ .





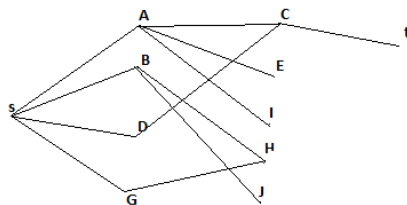
## NA1



- Sigue  $G$  en la cola, con vecinos  $A, H$ .

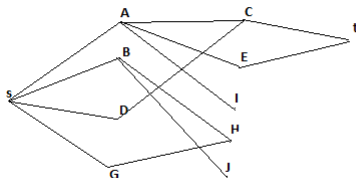
- $A, G \in V_1 \Rightarrow$  NO agregamos  $\overrightarrow{GA}$ .
- $G \in V_1, H \in V_2 \Rightarrow$  agregamos  $\overrightarrow{GH}$ .

## NA1



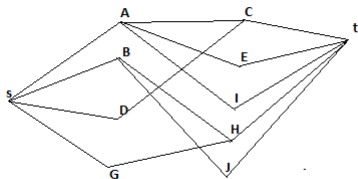
- Terminamos  $V_1$ , vamos con el primer vértice de  $V_2$ :  $C$ .
- $C$  tiene vecinos a  $t$ ,  $E$ ,  $G$ .
- $E$  está en el mismo nivel que  $C$  y  $G$  está en un nivel anterior, así que los lados  $\overrightarrow{CE}$ ,  $\overrightarrow{CG}$  no van.
- Creamos  $V_3 = \{t\}$  y agregamos  $\overrightarrow{Ct}$ .

## NA1



- Sigue  $E$  que tiene vecinos  $t, F$ .
  - Agregamos  $\overrightarrow{Et}$  al network.
  - Deberíamos agregar  $F$  a  $V_3$  y  $\overrightarrow{EF}$  al network,
  - pero el último nivel es siempre  $\{t\}$ , porque cualquier otro vértice que estuviese en ese mismo nivel, ya no podría llegar a  $t$  en ese network auxiliar, así que directamente no se incluye.

## NA1



- Finalmente agregamos los lados  $\overrightarrow{It}$ ,  $\overrightarrow{Ht}$ ,  $\overrightarrow{Jt}$  al network auxiliar.
- $V_0 = \{s\}$ ,  $V_1 = \{A, B, D, G\}$ ,  $V_2 = \{C, E, I, H, J\}$ ,  $V_3 = \{t\}$ .
- Lados  $\overrightarrow{sA}$ ,  $\overrightarrow{sB}$ ,  $\overrightarrow{sD}$ ,  $\overrightarrow{sG}$ ,  $\overrightarrow{AC}$ ,  $\overrightarrow{AE}$ ,  $\overrightarrow{AI}$ ,  $\overrightarrow{BH}$ ,  $\overrightarrow{BJ}$ ,  $\overrightarrow{DC}$ ,  $\overrightarrow{GH}$ ,  $\overrightarrow{Ct}$ ,  $\overrightarrow{Et}$ ,  $\overrightarrow{It}$ ,  $\overrightarrow{Ht}$ ,  $\overrightarrow{Jt}$ .

## NA1-1er camino

- Buscamos caminos dirigidos entre  $s$  y  $t$ .
- $sACt$  es un tal camino.
- Recordemos que en Edmonds-Karp podíamos ir calculando los  $\varepsilon(x)$  a medida que construíamos la cola, tomando el mínimo de lo que podíamos transportar por el lado y el  $\varepsilon$  del vértice que ponía a  $x$  en la cola.
- Pero aca, como estamos codificando en realidad múltiples posibles caminos, no podemos asignarle ningún  $\varepsilon(x)$  a los vértices.
- Así que el  $\varepsilon$  del camino hay que calcularlo luego de haberlo construido.

## NA1-1er camino

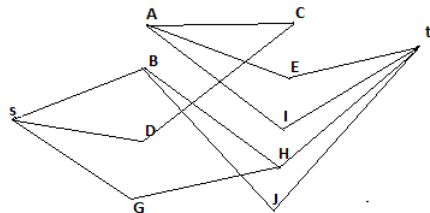
$\overrightarrow{sA} : 8, \overrightarrow{sB} : 7, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 9, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4$   
 $\overrightarrow{BH} : 9, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 8, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 7, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9$

- En este caso, como  $\overrightarrow{sA}$  y  $\overrightarrow{Ct}$  tienen capacidad 8 y  $\overrightarrow{AC}$  capacidad 9, podemos mandar 8 unidades de flujo por el camino  $sACt$ .
- Mandando 8 unidades de flujo por el camino  $sACt$ , actualizamos directamente sobre el network original, quedando:

## NA1-1er camino

- En este caso, como  $\overrightarrow{sA}$  y  $\overrightarrow{Ct}$  tienen capacidad 8 y  $\overrightarrow{AC}$  capacidad 9, podemos mandar 8 unidades de flujo por el camino  $sACt$ .
- Mandando 8 unidades de flujo por el camino  $sACt$ , actualizamos directamente sobre el network original, quedando:
- $\overrightarrow{sA} : 0, \overrightarrow{sB} : 7, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4$   
 $\overrightarrow{BH} : 9, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 7, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

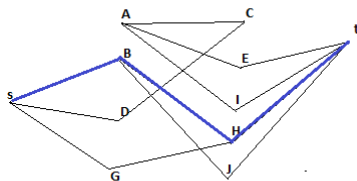
# NA1-1er camino



- En el network auxiliar, borramos los lados saturados.
- Podemos tacharlos (a mano)
- O aca, puedo efectivamente borrarlos.

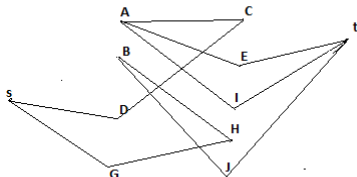


## NA1-2do camino



- Si hubieramos hecho Edmonds-Karp tendríamos que repetir buena parte del trabajo hecho antes, construyendo una cola, etc.
- En cambio aca simplemente miramos el network y buscamos un camino:
- saliendo por  $\overrightarrow{sB}$ , obtenemos el camino  $sBHt$ .

## NA1-2do camino



- Podemos mandar 7 unidades por ahí.
- Luego de ese aumento se saturan los lados  $\overrightarrow{sB}$  y  $\overrightarrow{Ht}$  así que los tachamos (a mano) y borramos (aquí).



## NA1

- En el network auxiliar, no podemos salir mas por ni por  $A$  ni por  $B$ .
- Saliendo por  $D$  podemos llegar hasta  $C$  pero luego no podemos seguir.
- Saliendo por  $G$ , podemos llegar hasta  $H$  pero luego no podemos seguir.
- Se termina la utilidad del network auxiliar 1, y hay que construir el 2do network auxiliar.

## NA2

- Para tenerlo a mano, los lados con  $c - f$  en cada lado:

- $\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- Las capacidades de los lados del network auxiliar serán esas, salvo en un lado del cual hablaré luego.
- Network auxiliar empieza con  $V_0 = \{s\}$ .

- $s$  agrega a  $D$  y  $G$ , (pues  $\overrightarrow{sA}, \overrightarrow{sB}$  están saturados) que forman  $V_1$  y se agregan al network auxiliar los lados  $\overrightarrow{sD}, \overrightarrow{sG}$ .

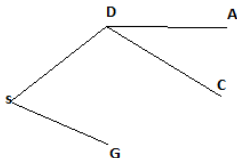
## NA2

$$\begin{aligned} \overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4 \\ \overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9 \\ \overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9. \end{aligned}$$

- Sigue  $D$  que tiene vecinos a  $A, C, G$ .
- $D$  agrega a  $A$  y  $C$ , creando el conjunto  $V_2$  con ellos, y se agregan los lados  $\overrightarrow{DA}, \overrightarrow{DC}$ .

## NA2

- Sigue  $D$  que tiene vecinos a  $A, C, G$ .
- $D$  agrega a  $A$  y  $C$ , creando el conjunto  $V_2$  con ellos, y se agregan los lados  $\overrightarrow{DA}, \overrightarrow{DC}$ .



- El lado  $\overrightarrow{DG}$  no se agrega pues ambos vertices están en el mismo nivel.

## NA2

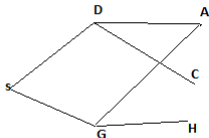
$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- $G$  tiene vecinos a  $A$ , que ya está, y a  $H$ , que no.



## NA2

- $G$  tiene vecinos a  $A$ , que ya está, y a  $H$ , que no.



- Como  $G \in V_1, A \in V_2$ , podemos agregar el lado  $\overrightarrow{GA}$ .
- Como  $H$  no está, lo agregamos a  $V_2$  y agregamos el lado  $\overrightarrow{GH}$  al network.

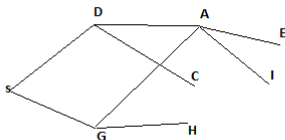
## NA2

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- Sigue A en la cola, con vecinos C, E, I.

## NA2

- Sigue  $A$  en la cola, con vecinos  $C, E, I$ .



- $C$  está en el mismo nivel que  $A$  ( $V_2$ ) así que no se agrega  $\overrightarrow{AC}$ .
- $E, I$  no están, así que se crea  $V_3$  con ellos y se agregan los lados  $\overrightarrow{AE}, \overrightarrow{AI}$ .

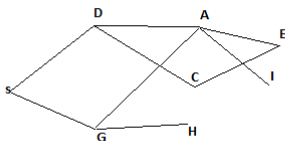
## NA2

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- El que sigue en la cola es  $C$ . Sus vecinos son  $E, G...$  y uno mas, que ya hablaremos. (No es  $t$ ,  $\overrightarrow{Ct}$  está saturado).

## NA2

- El que sigue en la cola es  $C$ . Sus vecinos son  $E, G$ ....y uno mas, que ya hablaremos. (No es  $t$ ,  $\overrightarrow{Ct}$  está saturado).



- $C \in V_2, E \in V_3$  se agrega el lado  $\overrightarrow{CE}$ .
- $C \in V_2, G \in V_1$  por lo tanto el lado  $\overrightarrow{CG}$  NO se agrega.
- Parece que terminamos con los vecinos de  $C$ , pero yo dije que habia “uno mas”

## NA2

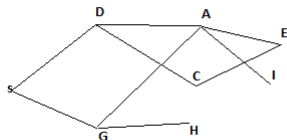
- Es A!
- Pues enviamos flujo por el lado  $\overrightarrow{AC}$  en el primer NA.

## NA2

- Es A!
- Pues enviamos flujo por el lado  $\overrightarrow{AC}$  en el primer NA.
- Podemos ver que la capacidad sobrante actual es 1 mientras que la original era 9, así que hay flujo de 8 por ese lado.  $\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9,$   
 $\overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4, \overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9,$   
 $\overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9, \overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9,$   
 $\overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

## NA2

- Es A!
- Pues enviamos flujo por el lado  $\overrightarrow{AC}$  en el primer NA.
- Podemos ver que la capacidad sobrante actual es 1 mientras que la original era 9, así que hay flujo de 8 por ese lado.
- así que ahora  $C$  puede devolverle flujo a  $A$ .
- Sólo que, como  $A, C$  están ambos en  $V_2$ , no hacemos nada.





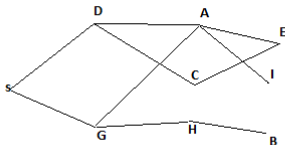
## NA2

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

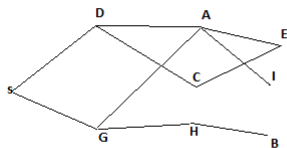
- Sigue  $H$ , pero su único vecino es  $t$  y  $Ht$  esta saturado.

## NA2

- Sigue  $H$ , pero su único vecino es  $t$  y  $Ht$  está saturado.
- Pero en realidad  $t$  no es su único vecino. Es su único vecino **hacia adelante**.
- Como  $B$  le mandó flujo a  $H$  en el primer NA, ahora  $H$  le puede devolver.
- Agregamos  $B$  a  $V_3$  y el lado  $\overrightarrow{HB}$  al network auxiliar.



## NA2



- Observemos que  $\overrightarrow{HB}$  no existe en el network original, pero si en el auxiliar.
- ¿Y cual es su capacidad?
- Lo que puede devolver, en este caso 7.

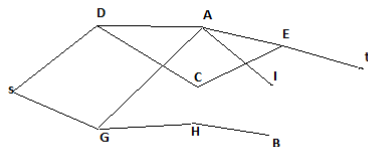
## NA2

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- $E$  tiene vecinos  $t, F$ . Se crea  $V_4 = \{t\}$ , el lado  $\overrightarrow{Et}$  y  $F$  se ignora pues tendría que estar en el mismo nivel que  $t$ .

## NA2

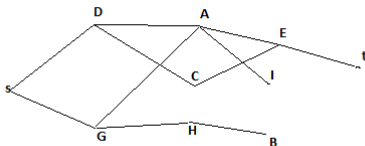
- $E$  tiene vecinos  $t, F$ . Se crea  $V_4 = \{t\}$ , el lado  $\overrightarrow{Et}$  y  $F$  se ignora pues tendría que estar en el mismo nivel que  $t$ .



## NA2

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

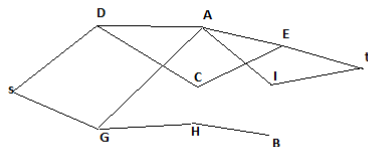
- $E$  tiene vecinos  $t, F$ . Se crea  $V_4 = \{t\}$ , el lado  $\overrightarrow{Et}$  y  $F$  se ignora pues tendría que estar en el mismo nivel que  $t$ .



- Sigue  $I$ , se agrega  $\overrightarrow{It}$ .

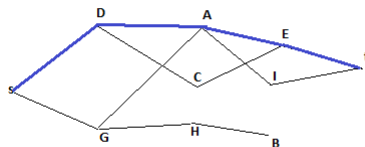
## NA2

- $E$  tiene vecinos  $t, F$ . Se crea  $V_4 = \{t\}$ , el lado  $\overrightarrow{Et}$  y  $F$  se ignora pues tendría que estar en el mismo nivel que  $t$ .



- Sigue  $I$ , se agrega  $\overrightarrow{It}$ .
- Sigue  $B$  con vecinos  $H$  (nivel anterior) y  $J$ , que no está.
- Pero como  $J$  tendría que ponerse en el mismo nivel que  $t$ , se ignora, así que terminamos la construcción.

## Ejemplo (cont.)



- En el algoritmo original de Dinitz en realidad ahora habría que hacer algo extra, pero hablaré de eso luego.
- Buscando caminos, encontramos  $sDAEt$ .



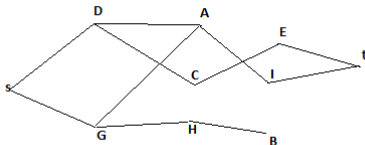
## Ejemplo (cont.)

$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 9, \overrightarrow{sG} : 9, \overrightarrow{AC} : 1, \overrightarrow{AE} : 1, \overrightarrow{AI} : 4,$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 1, \overrightarrow{CG} : 9, \overrightarrow{DA} : 2, \overrightarrow{DC} : 9, \overrightarrow{DG} : 9,$   
 $\overrightarrow{Et} : 9, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 1, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 9, \overrightarrow{Jt} : 9.$

- En el algoritmo original de Dinitz en realidad ahora habria que hacer algo extra, pero hablaré de eso luego.
- Buscando caminos, encontramos  $sDAEt$ .
- Podemos mandar 1 una unidad de flujo, saturando  $\overrightarrow{AE}$ .

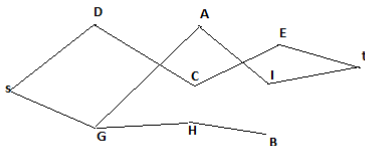
## Ejemplo (cont.)

- En el algoritmo original de Dinitz en realidad ahora habria que hacer algo extra, pero hablaré de eso luego.
- Buscando caminos, encontramos  $sDAEt$ .
- Podemos mandar 1 una unidad de flujo, saturando  $\overrightarrow{AE}$ .



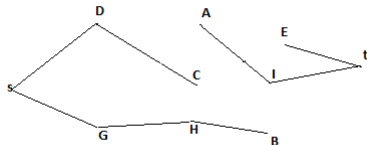
## Ejemplo (cont.)

- Deberia actualizar las capacidades sobrantes del network, pero para no escribir tanto, se los dejo como ejercicio. (lo haré al final del network auxiliar).
- Luego encontramos  $sDA/t$



- Se puede mandar 1, se satura  $\overrightarrow{DA}$ .

## Ejemplo (cont.)

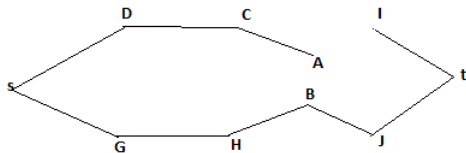


- Seguimos con  $sDCEt$ .
- Podemos mandar también 1, saturando  $\overrightarrow{CE}$ .
- Siguiendo camino es  $sGAIt$ , también mandamos 1 y se satura  $GA$ .
- No quedan más caminos entre  $s$  y  $t$ .

## Ejemplo (cont.)

- Luego de todos esos caminos  $c - f$  queda:(en azul todos los que cambiaron)
- $\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 6, \overrightarrow{sG} : 8, \overrightarrow{AC} : 1, \overrightarrow{AE} : 0, \overrightarrow{AI} : 2$   
 $\overrightarrow{BH} : 2, \overrightarrow{BJ} : 9, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 0, \overrightarrow{CG} : 9, \overrightarrow{DA} : 0, \overrightarrow{DC} : 8, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 7, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 0, \overrightarrow{GH} : 9, \overrightarrow{Ht} : 0, \overrightarrow{It} : 7, \overrightarrow{Jt} : 9.$
- La construcción del tercer network auxiliar no la voy a describir tan en detalle como hice con esta.
- Además de que es mucho más fácil. Si uno mira la tabla anterior y construye el tercer network auxiliar, obtiene:

## NA3



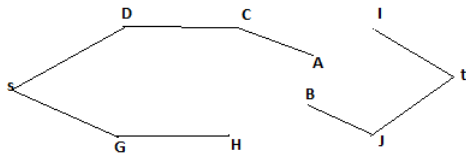
■ Tenemos dos caminos:

1  $sDCAIt : 2$ , que corresponde al camino  $sDCAIt : 2$  del network original.

■ Se satura el lado  $AI$ .

2 y  $sGHBjt : 7$  que corresponde en el network original a  $sGHB.It : 7$

## NA3



- Sólo podemos mandar 7 pues  $H$  sólo le puede devolver 7 a  $B$ .
- Es decir, el lado  $\overrightarrow{HB}$  en el network auxiliar se satura.
- Corresponde a que el lado  $\overrightarrow{BH}$  en el network original se vacía completamente.

## NA3

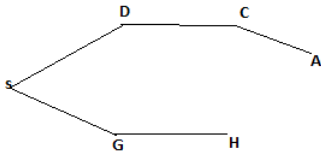
- $s$  queda desconectado de  $t$ , terminamos con el tercer network auxiliar
- Ahora doy la actualización completa del flujo luego de los dos caminos de este network auxiliar.
- Al igual que en EK, en los lados backward  $c - f$  aumenta.
- Esos los voy a marcar en rojo, los otros que cambiaron en azul.
- $\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 4, \overrightarrow{sG} : 1, \overrightarrow{AC} : 3, \overrightarrow{AE} : 0, \overrightarrow{AI} : 0$   
 $\overrightarrow{BH} : 9, \overrightarrow{BJ} : 2, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 0, \overrightarrow{CG} : 9, \overrightarrow{DA} : 0, \overrightarrow{DC} : 6, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 7, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 0, \overrightarrow{GH} : 2, \overrightarrow{Ht} : 0, \overrightarrow{It} : 5, \overrightarrow{Jt} : 2.$



## NA4

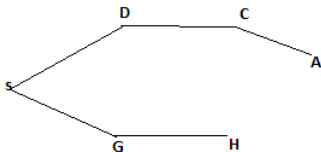
$\overrightarrow{sA} : 0, \overrightarrow{sB} : 0, \overrightarrow{sD} : 4, \overrightarrow{sG} : 1, \overrightarrow{AC} : 3, \overrightarrow{AE} : 0, \overrightarrow{AI} : 0$   
 $\overrightarrow{BH} : 9, \overrightarrow{BJ} : 2, \overrightarrow{Ct} : 0, \overrightarrow{CE} : 0, \overrightarrow{CG} : 9, \overrightarrow{DA} : 0, \overrightarrow{DC} : 6, \overrightarrow{DG} : 9$   
 $\overrightarrow{Et} : 7, \overrightarrow{EF} : 9, \overrightarrow{Ft} : 9, \overrightarrow{GA} : 0, \overrightarrow{GH} : 2, \overrightarrow{Ht} : 0, \overrightarrow{It} : 5, \overrightarrow{Jt} : 2.$

- Si ahora construimos el cuarto network auxiliar, obtenemos:



## NA4

- Si ahora construimos el cuarto network auxiliar, obtenemos:



- No pudimos llegar a  $t$ .
- El flujo es maximal. Un corte minimal es:
- $S = \text{vértices del último network auxiliar}$   
 $= \{s, D, G, C, H, A\}$ .

## Ejemplo

- Valor del flujo:  $f(\overrightarrow{sA}) + f(\overrightarrow{sB}) + f(\overrightarrow{sD}) + f(\overrightarrow{sG}) = 8 + 7 + (9 - 4) + (9 - 1) = 15 + 5 + 8 = 28$ .
- Capacidad del corte  $\{s, D, G, C, H, A\}$
- $c(\overrightarrow{sB}) + c(\overrightarrow{Ct}) + c(\overrightarrow{CE}) + c(\overrightarrow{Ht}) + c(\overrightarrow{AE}) + c(\overrightarrow{AI})$
- $= 7 + 8 + 1 + 7 + 1 + 4 = 28$ .
- Hemos chequeado que  $f$  es maximal y  $S$  minimal.



## Observaciones

- Podemos ver en el ejemplo que el primer NA tenía 4 niveles, el 2do NA tenía 5 y el tercero 6.
- (el cuarto tenía sólo 4, pero no llegabamos a  $t$ ).
- Esto pasa siempre: un network auxiliar “codifica” TODOS los caminos de una longitud dada.
- Cuando terminemos con ese NA, todos los demas caminos que encontremos van a tener longitud estrictamente mayor.
- Esto es algo que probaremos y es clave para la complejidad de todos los algoritmos “tipo” Dinic.

## Observaciones

- Al igual que el punto anterior, esto también les sirve cuando lo corran a mano para darse cuenta si hicieron algo espantosamente mal.
- Si al correr el algoritmo su network auxiliar (que no sea el último, el que no llega a  $t$ ) tiene **menos** o incluso **la misma cantidad** de niveles que alguno anterior, entonces hicieron algo mal.
- Al igual que con el punto anterior, como esto es algo fácil de verificar, si no lo hacen les descontaremos muchos puntos.