Demostraciones:

3) Complejidad algoritmo Dinic-original y Dinic-even.

- La d(s,t) entre NA consecutivas aumenta, hay a lo sumo n NA
- Sea CFB la complejidad de encontrar flujo bloqueante en un NA.

un algoritmo "tipo" Dinic tiene complejidad.

$(N° NA's) * (C_{crear} NA + CFB) = O(n*(m+CFB))$

hay que ver que CFB $O(mn)$

Original

- cada camino entre s y t se encuentra con DFS, pero se asegura de NO hacer backtracking (vertice con lado entrante tiene lado saliente)
- para mantener esto se debe ir revisando la NA, "Podar"
  - debemos encontrar complejidad de: encontrar caminos + podar.

Construir un camino dirigido desde s a t es muy fácil:

- Tomar $p = s$.
- WHILE($p \neq t$)
  - Tomar q cualquier vécino de p. (*)
  - Agregar $\overrightarrow{pq}$ al camino.
  - Tomar $p = q$
- ENDWHILE

la construccion depende de la cantidad de niveles, entonces es $O(n)$. Luego se borran los lados saturados en el network $O(n)$ de vuelta.

- como cada camino satura y por lo tanto borra al menos un lado, hay a lo sumo m caminos.

complejidad encontrar caminos = $O(mn)$

Complejidad PODAR:

podar consiste en recorrer los vertices y eliminar los que no tienen salida.

- PV: recorrer vertices = $O(n)$ chequea todos los vertices.
- B(x): eliminar lados de entrada de un vertice X. } una sola vez por vertice.

hay un PV en cada PODAR (1 podar por camino) a lo sumo m+1 PV.

complejidad todos PV $O(nm)$

Complejidad $B(x) = O(d(x))$, y se hace una sola vez por x.
complejidad del conjunto $B(x) = O(\sum_x d(x)) = O(2m) = O(m)$
Complejidad CFB:
encontrar-caminar + podar
$O(nm) + O(PV) + O(B(x)) = O(nm) + O(nm) + O(m)$
$= O(nm)$

Complejidad Dinic-Original $= O(n*(m+nm))$
$= O(mn^2)$

Dinic-Even

-No tiene podar, ahora DFS puede tener que hacer backtrack.

Tenemos que ver el pseudocodigo con el que se encuentran flujos bloqueante en una NA.

```
g = 0
STOPFLAG:=1//para saber cuando parar
WHILE (STOPFLAG) //while externo
|   p = [s], x = s // Inicialización inicial de x y del camino p
|   WHILE ((x ≠ t) AND (STOPFLAG))//while interno
|   |   IF Γ+(x) ≠ ∅ THEN AVANZAR(x)
|   |       ELSE IF (x ≠ s) THEN RETROCEDER(x)
|   |           ELSE STOPFLAG=0
|   IF (x == t) THEN INCREMENTAR
RETURN(g)
```

avanzar y retroceder, $O(1)$.
Incrementar $O(n)$, largo max del camino en N e incrementar aumenta el flujo a lo largo del camino y borra los saturados.

Si denotamos AVANZAR por A, RETROCEDER por R e INCREMENTAR+inicializar por I, entonces vemos que Dinic-Even es una sucesión de As,Rs,Is.

podemos dividir la suc de palabras en
$AA\ldots A\underbrace{X}_{R\,o\,I}$

Cada palabra termina en R o I, y tanto R o I borran al menos un lado, cant de palabras a lo sumo m.

Complejidad de cada palabra:
R, A $O(1)$, I $O(n)$, si la palabra $AA\ldots X$ tiene k As, la complejidad sera $O(k+1) = O(k)$ o $O(k+n) = O(n)$, pero como cada A significa aumentar 1 nivel en el NA, entonces $k \leq n$, por lo que sea $X = R\,o\,I$ la complejidad de las palabras es $O(n)$, y como hay a lo sumo m palabras, entonces la complejidad de encontrar un flujo bloqueante es $O(nm)$ y la de Dinic-Even $O(mn^2)$

4) Complejidad Wave:

- Sabemos que hay a lo sumo n NA y la cantidad de niveles aumenta.
- hay que ver que la complejidad de encontrar un flujo bloqueante en una NA es $O(n^2)$ entonces la la complejidad de wave es $O(n) \cdot O(n^2) = O(n^3)$.

Prueba:

en Wave hacemos serie de olas hacia delante y hacia atras.

- hacia adelante hacemos series de FB(x)
- hacia atras hacemos serie de BB(x)

por cada uno revisamos $\Gamma^+(x)$ o $M(x)$.

en gral cuando miramos un $y \in \Gamma^+(x)$ pueden pasar 2 cosas:

1. $\overrightarrow{xy}$ se satura, S cant total sobre todas las olas hacia adelante.
2. no se satura, P cant total sobre todas las olas hacia adelante.

Cuando miramos $y \in M(x)$ pasan dos cosas 1. $\overrightarrow{xy}$ se vacia 2. no se vacia.

Sea *V* la cantidad total sobre todas las olas hacia atras, de procesamientos de la categoria [I] arriba, y *Q* la cantidad total de procesamientos sobre todas las olas hacia atras, de lados de la categoria [II] arriba.

la complejidad del paso bloqueante es $S + P + V + Q$.

S.

- supongamos que un lado $\overrightarrow{xy}$ se satura entonces lo borrabamos del NA.

como nunca mas lo agregamos, nunca mas lo saturamos. (se saturan una vez)

S acotado por m.

nos aseguramos de que es correcto borrarlo porque para poder usarlo otra vez primero debe des-saturarse, y le devuelve flujo a x.

y solo le puede devolver flujo a x solo si y esta bloqueado, pero si esta bloqueado x no puede mandarle mas flujo a y, entonces $\overrightarrow{xy}$ no puede volver a ser usado.

V.

supongamos que $\overrightarrow{yx}$ se vacia.

Pero $\overrightarrow{yx}$ sólo se puede vaciar si *x* esta bloqueado pues de otra forma no podria devolverle flujo a *y*.

Pero si *x* esta bloqueado, el vértice "*y*" nunca mas puede mandarle flujo.

Si $\overrightarrow{yx}$ nunca mas puede recibir flujo, entonces menos aún va a poder volver a vaciarse.

Asi que *V* también está acotado por el número total de lados, *m*.

P

En cada $FB(x)$ buscamos vecinos de x y les mandamos todo el flujo que podamos: $Min\{D(x), c(\overrightarrow{xy}) - g(\overrightarrow{xy})\}$.

Si ese mínimo es igual a $c(\overrightarrow{xy}) - g(\overrightarrow{xy})$ el lado $\overrightarrow{xy}$ queda saturado, lo retiramos de $\Gamma^+(x)$ y continuamos con otro.

Entonces, de entre todos los vécinos de x hay A LO SUMO uno sólo tal que ese mínimo es $D(x)$.

Es decir, al hacer $FB(x)$, todos los lados $\overrightarrow{xy}$ que miramos, salvo a lo sumo uno, se saturan.

Concluimos entonces que en cada FB hay a lo sumo UN lado procesado como parte de P.

por lo que decimos que P esta acotado superiormente, por la cant de FB.

en cada ola hay a lo sumo $n-2$ FB.

Cuantas olas hacia delante hay?

en cada ola menos la ultima al menos un vertice se bloquea, si no bloqueo, entonces balanceo a todos y es la ultima, entonces hay a lo sumo n olas hacia adelante.

Por lo tanto, tenemos a lo sumo $n-2$ FB por cada ola hacia adelante, y tenemos $O(n)$ olas hacia adelante, significa que tenemos en total $O(n^2)$ FB y por lo tanto, la cantidad de procesamientos de P es $O(n^2)$.

Similarmente, al hacer un BB, a lo sumo un lado no se vacia, asi que Q es la cantidad total de BB que son $n-2$ por cada ola hacia atras.

y como el número de olas hacia atras es igual al número de olas hacia adelante,

y vimos que estas estan acotadas por n,

tenemos que Q tambien es $O(n^2)$.

entonces la complejidad del paso bloqueante es

$$S+P+V+Q = O(m) + O(n^2) + O(m) + O(n^2) = O(n^2).$$

implica que la complejidad de wave es $O(n^3)$

5) La distancia en NA sucesivos aumenta

la cantidad de niveles de un NA es menor a la cantidad de niveles de un network posterior.

Prueba:

- Sea NA un network auxiliar y NA' un network sucesivo.
- Sea f el flujo (del network original) inmediatamente anterior a NA y f' el inmediatamente anterior a NA'.
- Sea $d(x) = d_f(s,x)$ y $d'(x) = d_{f'}(s,x)$, sabemos que $d(t) \leq d'(t)$ por la prueba de EK.

Queremos ver que vale a $<$.

Si $t \notin NA' \Rightarrow d'(t) = \infty > d(t)$ y ya esta, asumimos $t \in NA'$

$\Rightarrow \exists$ C. dirigido $x_0, x_1 \dots x_r$ entre s y t en NA', pero no puede ser camino en NA.

- Si lo fuera, al construir f' habriamos saturado ese camino. (es decir, saturar al menos un lado).
- Pues para terminar con NA y pasar de f a f' debemos saturar todos los caminos de NA.
- Pero si quedó saturado, no podria ser un camino en NA'.

Si el C-dirigido no está en NA se debe a 2 cosas.

1. algún x_i no está en NA.

2. están todos los x_i pero no algún lado $\overrightarrow{x_i x_{i+1}}$.

1.

Como $t \in NA \Rightarrow x_i \neq t$ y todos los vértices $\neq t$ que estén a distancia mayor o igual que t no se incluyen, pero todos los que tienen menor están pues NA se construye con BFS.

- la única forma que x_i no esté en NA es que $d(t) \leq d(x_i)$ (1)

y como sabemos que $d \leq d' \Rightarrow d(x_i) \leq d'(x_i)$ (2)

Como $x_1 x_2 \ldots x_{r-1} x_r$ es un camino en NA' y NA' es un net por niveles, concluimos que:

- $d'(x_i) = i \quad \forall i$ (3), además $x_i \neq t \Rightarrow i < r$ (4)

entonces.

$$d(t) \overset{(1)}{\leq} d(x_i) \overset{(2)}{\leq} d'(x_i) \overset{(3)}{=} i \overset{(4)}{<} r = d'(t)$$

y probamos que la distancia entre s y t aumenta para el caso 1.

2.

todos los $x_i \in NA$ pero $\exists\ \overrightarrow{x_i x_{i+1}} \notin NA$.

- sabemos que $d(x_{i+1}) \leq d'(x_{i+1})$, tenemos 2 subcasos que sea $<$ o $=$.

- $d(x_{i+1}) < d'(x_{i+1})$ (5), sea $b(x) = b_F(x,t)$, $b'(x) = b_{F'}(x,t)$

por EK sabemos $b \leq b'$ y $d(t) = d(x) + b(x)$, igual para d' y b'

$$\begin{aligned} d(t) &= d(x_{i+1}) + b(x_{i+1}) \\ &\leq d(x_{i+1}) + b'(x_{i+1}) \quad (E\text{-}K) \\ &\overset{(5)}{<} d'(x_{i+1}) + b'(x_{i+1}) = d'(t) \end{aligned}$$

, probamos $d(t) < d'(t)$

- Supongamos $d(x_{i+1}) = d'(x_{i+1}) = i+1$, como i es el primer i para el cual $\overrightarrow{x_i x_{i+1}} \notin NA$.

entonces el camino $x_0 x_1 \ldots x_i$ sí está en NA. $\Rightarrow d(x_i) = i$

entonces, en NA, x_i está en nivel i y x_{i+1} en nivel $i+1$

podemos concluir que ambos lados $\overrightarrow{x_i x_{i+1}}$ y $\overrightarrow{x_{i+1} x_i}$ no están en NA.

Como $d(x_i) = i$, $d(x_{i+1}) = i+1$. entonces x_i, x_{i+1} están a distancia "legal" para que exista el lado $\overrightarrow{x_i x_{i+1}}$

- Pero ese lado no esta en NA. Por lo que concluimos que:
  1. $\overrightarrow{x_i x_{i+1}}$ es lado del network original pero esta saturado, o:
  2. $\overrightarrow{x_{i+1} x_i}$ es lado del network original pero esta vacio.
- Pero $\overrightarrow{x_i x_{i+1}}$ si es un lado en NA'.
- Asi que la situación es:
  1. $\overrightarrow{x_i x_{i+1}}$ es lado del network original, estaba saturado al construir NA pero des-saturado al construir NA', o
  2. $\overrightarrow{x_{i+1} x_i}$ es lado del network original, estaba vacio al construir NA pero no vacio al construir NA'.

La única forma en que pase [1] es que al pasar de f a f' dessaturamos el lado $\overrightarrow{x_i x_{i+1}}$, es decir, devolvimos flujo, lo que dice que usamos en algún momento el lado como backward.

Como pasamos de f a f' usando NA, esto significa que en NA debemos haber usado el $\overrightarrow{x_{i+1} x_i}$.

Esto es un absurdo pues habiamos visto que ese lado no puede estar en NA, pues estariamos yendo del nivel $i+1$ al i.

La única forma en que pase [2] es que al pasar de f a f' mandamos algo de flujo por el lado $\overrightarrow{x_{i+1} x_i}$.

Asi que ese lado también debe ser un lado en NA, lo cual como dijimos es un absurdo.

Fin prueba ■

6) El valor de todo flujo es menor o igual que la capacidad de todo corte y que si f es un flujo, entonces las sig afirmaciones son eq:
i) $\exists$ corte S tq $v(f) = cap(S)$ y en este caso S es minimal
ii) f es maximal
iii) no existen f-caminos aumentantes.

Primero probamos $v(f) \leq cap(S)$, $\forall f, S$: f flujo y S corte. Usando que $v(f) = f(S,\bar{S}) - f(\bar{S},S)$.

luego probamos $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1$

Prueba:

- $v(f) \leq cap(S)$, $\forall S$ corte

$$\begin{aligned} v(f) &= f(S,\bar{S}) - f(\bar{S},S) \\ &\leq f(S,\bar{S}) \\ &\leq c(S,\bar{S}) \\ &= cap(S). \end{aligned}$$

$\therefore v(f) \leq cap(S)$

$1 \Rightarrow 2$. Sean f, S como en 1 y g un flujo cualq.
sabemos que $v(g) \leq cap(S)$ $\forall S$, entonces
$v(g) \leq cap(S) = v(f)$ para cualquier flujo g
entonces f es maximal, tambien
sea T un corte, sabemos $cap(T) \geq v(f) = cap(S)$
entonces S es minimal.

2 ⟹ 3 asumimos 2 veamos 3.

Sea f un flujo maximal, si existiese un f-camino aumentante, entonces podríamos mandar $\varepsilon > 0$ a través de él, y obtendríamos un flujo f' tq $v(f') = v(f) + \varepsilon > v(f)$ pero es absurdo, pues f es maximal por hipótesis.

3 ⟹ 1

no existen f-caminos aumentantes, entonces existe un corte S tq $cap(S) = v(f)$.

definimos un S como: $S = s \cup \{x \in V : \exists\ f\text{-Ca de } s \text{ a } x\}$

como vale 3 $\Rightarrow t \notin S$, S es un corte.

tenemos que $v(f) = f(S, \bar{S}) - f(\bar{S}, S)$

calculemos

$f(S, \bar{S}) = \sum f(\vec{xy})[x \in S][y \notin S][\vec{xy} \in E]$, vemos que el f-Ca de s a x existe pero a y no, eso significa que el lado $\vec{xy}$ está saturado, entonces $f(\vec{xy}) = c(\vec{xy})$

$= \sum c(\vec{xy}) = c(S, \bar{S}) = Cap(S)$

$f(\bar{S}, S) = \sum f(\vec{xy})[x \notin S][y \in S][\vec{xy} \in E]$, vemos que como $y \in S$ y $x \notin S$, el camino $S = x_0, x_1 \ldots x_r = y, x$ no es aumentante, pero podría serlo si se usara el lado $\vec{xy}$ como Backward, pero si no lo usamos, es porque $f(\vec{xy}) = 0$, entonces.

$f(\bar{S}, S) = \sum 0 = 0$

Luego $v(f) = Cap(S) - 0$

$v(f) = Cap(S)$ $\blacksquare$

Fin prueba y queda demostrado que las tres afirmaciones son equivalentes.

7) Probar que 2-Color es polinomial.