

1. Dado un digrafo D con pesos $c: E(D) \rightarrow \mathbb{N}$ y dos v rtices s y t , decimos que una arista $v \rightarrow w$ es *st-eficiente* cuando $v \rightarrow w$ pertenece a alg n camino m nimo de s a t . Sea $d(\cdot, \cdot)$ la funci n que indica el peso de un camino m nimo entre dos v rtices.

- Demostrar que $v \rightarrow w$ es *st-eficiente* si y s lo si $d(s, v) + c(v \rightarrow w) + d(w, t) = d(s, t)$.
- Usando el inciso anterior, proponga un algoritmo eficiente que encuentre el m nimo de los caminos entre s y t que no use aristas *st-eficientes*. Si dicho camino no existe, el algoritmo retorna \perp .

$$\textcircled{a} \quad v \rightarrow w \text{ es st-eficiente} \iff d(s, v) + c(v \rightarrow w) + d(w, t) = d(s, t)$$

\rightarrow) Sabemos que $v \rightarrow w$ es *st-eficiente*
 $\forall v, w \quad d(s, v) + c(v \rightarrow w) + d(w, t) = d(s, t)$

Dado $v \rightarrow w$ *st-eficiente* sabemos que $v \rightarrow w \in P_{s \rightarrow t}$
 con $P = \delta(s, t) \rightarrow P$ es una uni n de caminos m nimos
 los subcamino
 de un camino
 son m nimos

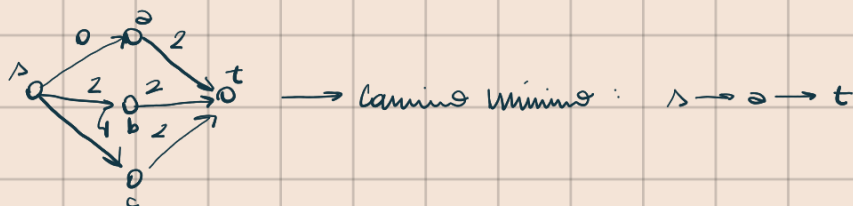
$$\rightarrow \sum_{\substack{x \rightarrow y \\ \in P}} d(x, y) = d(s, t) \rightarrow \text{en particular } d(s, v) + c(v \rightarrow w) + d(w, t) = d(s, t)$$

\leftarrow) Sea $d(s, t) = d(s, v) + c(v \rightarrow w) + d(w, t)$ $\forall v, w$ $v \rightarrow w$ es *st-eficiente*

Sup que $v \rightarrow w$ NO es *st-eficiente* $\rightarrow v \rightarrow w$ NO pertenece a ning n
 camino m nimo $\rightarrow d(s, v) + c(v \rightarrow w) + d(w, t) > d(s, t)$
 !ABS!

⑥ Queremos el mínimo de los caminos entre s y t que NO use aristas st-eficientes

Veamos un ejemplo



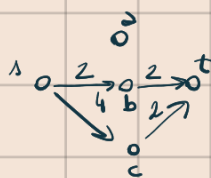
→ por el inciso a sabemos que $s \rightarrow a$ y $a \rightarrow t$ son st-eficientes

$$\rightarrow d(s, s) + c(s, a) + d(a, t) = d(s, t)$$

idem para $a \rightarrow t$

→ Vamos a querer que el algo derecho $s \rightarrow b \rightarrow t$

Es decir, vamos a trabajar sobre



Vamos a necesitar identificar las aristas st-eficientes, quitarlas y hacer Dijkstra
 ↳ la única forma de identificar las st-eficientes es hacer Dijkstra

Algoritmo

1. Hacer Dijkstra en G $O(\min\{m \log n, n^2\})$
2. Eliminar las aristas candidatas \rightarrow recorrer de $u \rightarrow v$ y los guardo
 $O(n) \rightarrow \forall e \in E(G) \quad d(s, u) + c(e) + d(v, t) = d(s, t) \rightarrow$ hacer esto
3. Considero $H = (V(G), E(G) \setminus \text{aristas st-eficientes})$
4. Hago Dijkstra en H
5. Si $\exists p_H s \rightarrow t$ lo devuelvo si no devuelvo \perp . He podido fijar si alguno NO está en el árbol