14. Dado un digrafo D con pesos $c\colon E(D) \to \mathbb{R}$ que no tiene ciclos de peso negativo, queremos encontrar la arista $v \to w$ que sea st-eficiente para la mayor cantidad de pares s y t. Proponer un algoritmo eficiente y "simple de programar" para resolver este problema. **Ayuda:** verificar que la propiedad del Ejercicio 1a también es cierta en este caso.

Queremos la arista $v \to w$ que esté en la mayor # de caminos mínimos entre todos los pares de vértices

Algoritmo

1. Hago FW sobre D y pido que devuelva MCM P $\quad O(n^3)$
   Voy a tener n v-ACMs, 1 por cada $v \in V(G)$
2. Para c/ v-ACM marco las aristas. Las que se repitan con 1 mismo color
3. Me fijo cuál es la arista pintada más veces y devuelvo esa arista

Más implementativo

1. FW sobre D y me quedo con MCM $\quad O(n^3)$
2. Inicializo un diccionario $H: e \longrightarrow \mathbb{N}$ $\quad O(L)$
3. Recorro cada v-ACM y para c/ (i,j) lo agrego a H si no estaba y si estaba aumento su significado $\quad O(n^2)$
4. Recorro H y devuelvo $(\hat{i},\hat{j}) / H(\hat{i},\hat{j}) = \max\{H(i,j) / (i,j) \in \text{clave } H\}$ $\quad O(n)$

¿Este enfoque sería correcto si existiese un único camino mínimo de v a w
pero podría NO ser el caso

1. Hago FW sobre D
2. Inicializo un diccionario H: aristas $\to \mathbb{N}$
3. Para cada $(s,t) \in V(G)$
4.   Para cada $(v,w) \in E(G)$
5.   Si $d(s,v)+c(v,w)+d(w,t) = d(s,t)$
6.   La agrego a H si no estaba con significado 1 o
7.   aumento su significado
8. Recorro el diccionario en busca de la arista de mayor significado