6. Dado un digrafo D con pesos $c\colon E(D) \to \mathbb{N}$ y dos vértices s y t, decimos que una arista $v \to w$ es *st-eficiente* cuando $v \to w$ pertenece a algún camino mínimo de s a t. Sea $d(\cdot,\cdot)$ la función que indica el peso de un camino mínimo entre dos vértices.

---

a. Demostrar que $v \to w$ es st-eficiente si y sólo si $d(s,v) + c(v \to w) + d(w,t) = d(s,t)$.

Teorema $v \to w$ es *st*-eficiente $\iff d(s,v) + c(v \to w) + d(w,t) = d(s,t)$

$(\Longrightarrow)$

Hipotesis: Sabemos que $v \to w$ existe en algún camino mínimo de *s* a *t* con $d(s,t)$

Y por propiedad del camino más corto sabemos que $d(s,v) + d(v,w) + d(w,t)$, y sabemos que $v \to w$ está en un camino mínimo de *s* a *t* por hipótesis, por lo que $d(v,w) = c(v \to w)$ tal que

$d(s,v) + c(v \to w) + d(w,t) = d(s,v) + d(v,w) + d(w,t) = d(s,t)$

$(\Longleftarrow)$

Hipotesis: $d(s,v) + c(v \to w) + d(w,t) = d(s,t)$

Si $d(s,v) + c(v \to w) + d(w,t) = d(s,t)$ entonces $v \to w$ existe en un camino mínimo de *s* a *t*, por lo que es *st*-eficiente.

---

b. Usando el inciso anterior, proponga un algoritmo eficiente que encuentre el mínimo de los caminos entre s y t que no use aristas st-eficientes. Si dicho camino no existe, el algoritmo retorna $\perp$.

```
Asumo G como lista de adyacencia.

Algoritmo(G):
  ds <- Dijkstra(G,s) que devuelve los caminos minimos desde s a todos

  G' <- G con las aristas invertidas
  dt <- Dijkstra(G',t) que devuelve los caminos minimos desde t a todos

  Gout <- conjunto vacio

  para cada v en G:
    para cada w en G[v]:
      si ds(v) + c(v->w) + dt(w) != ds(t):
        agrego (v,w) a Gout

  res <- Dijkstra Gout

  si res es vacio:
    return ⊥
  sino:
    return res
```

La idea del algoritmo es ver que hay camino tal que no hay arista $c(v \to w) = d(v,w)$ y necesariamente haya un camino $P = v...w$ diferente a $v \to w$ que sea mínimo.