3. Diseñar un algoritmo eficiente que, dado un digrafo pesado G y dos vértices s y t, determine el recorrido minimo de s a t que pasa por a lo sumo una arista de peso negativo. **Demostrar** que el algoritmo propuesto es correcto.

El recorrido mínimo de s a t debe tener 0 o 1 arista negativa

Algoritmo

1. Armar el grafo $H = (V(G), E(G) \setminus \{v \to w \in E(G) / c(v \to w) < 0\})$
2. Hacer Dijkstra en H desde s y hacerlo también en H^t desde t
3. Para cada $x \to y \in \{x \to y \in E(G) \wedge c(x \to y) < 0\}$ calcular $d(s,x) + c(x \to y) + d(y,t)$
4. Devolver el camino de menor peso de s a t de entre los que tienen una arista negativa y el que NO si es que existe

Dem

Como queremos los recorridos con a lo sumo una arista negativa, en particular vamos a querer los caminos ya que los recorridos positivos van a tener un peso mayor que los caminos porque pueden repetir aristas.
Entonces si a G le sacamos las aristas negativas, generando H, y hacemos Dijkstra desde s en H y desde t en H^t vamos a obtener los caminos mínimos de s a todos los vértices y de t al resto de vértices.
De esta manera, si vamos agregando las aristas negativas podemos aprovechar que tenemos los caminos mínimos de peso positivo desde s y t y fijarnos cuánto más chico se vuelve el valor del ahora recorrido.

Veamos que costo T devuelto por el algo - costo de T que resuelve el problema

$$c_+(T') = \min c_+(T)$$

→ Como $\min c_+(T) =$ es el costo mínimo de los recorridos en G con al menos una arista negativa por def $\min c_+(T) \leq c_+(\tilde{T}) \ \forall \tilde{T}'$ recorrido con al menos 1 arista negativa

$\longrightarrow \min C_+(T) \leq C_+(T')$

$\longrightarrow$ $C_+(T') = d(s,x) + c(x \to y) + d(y,t)$ con $d(s,x)$ y $d(y,t)$

Camino mínimo $\to C_+(T') \leq C_+(\tilde{T})$ $\forall \tilde{T}$ recorrido con al menos 1 arista negativa $\longrightarrow C_+(T') \leq \min C_+(\tilde{T})$

$C_+(T') \leq \min C_+(T)$