5. Sea G un digrafo con pesos positivos que tiene dos vértices especiales s y t. Decimos que una arista $e \in E(G)$ es *crítica* para s y t cuando $d_G(s,t) < d_{G-e}(s,t)$. Diseñar un algoritmo eficiente que, dado G, determine las aristas de G que son críticas para s y t. **Demostrar** que el algoritmo es correcto. **Ayuda:** pensar en el subgrafo P de G que está formado por las aristas de caminos mínimos de G (el "grafo de caminos mínimos").

Las aristas críticas son las aristas puentes de P. Al sacar estas aristas, por ser puentes en P, no existe camino entre s y t en P, y de existir camino en G, éste no sería mínimo pues sino debería haber estado en P.

1) Sea P = DAG de caminos mínimos entre s y t con Dijkstra.
$O(m \lg n)$

2) Buscamos todas las aristas puente en P.
$O(m+n)$

$e=(u,v)$ arista puente en P (DAG de caminos mínimos en G desde s)

$\iff$ e es una arista crítica de G: $d_G(s,t) < d_{G-e}(s,t)$

($\Rightarrow$)

Supongamos que e no es crítica. Entonces $d_G(s,t) \geq d_{G-e}(s,t)$. Notemos que no puede suceder $d_G(s,t) > d_{G-e}(s,t)$ pues $d_G(s,t)$ es el camino mínimo entre s y t en G, sacar una arista nunca puede mejorar el camino mínimo. Luego $d_G(s,t) = d_{G-e}(s,t)$, y como e no es crítica existe un camino mínimo entre s y t que no pasa por e. Por lo tanto e no sería puente en P. Absurdo porque e es puente en P por hipótesis. Luego $d_G(s,t) < d_{G-e}(s,t)$ y por lo tanto e es una arista crítica.

($\Leftarrow$)