

2. Diseñar un algoritmo eficiente que, dado un digrafo G con pesos no negativos, dos vértices s y t y una cota c , determine una arista de peso máximo de entre aquellas que se encuentran en algún recorrido de s a t cuyo peso (del recorrido, no de la arista) sea a lo sumo c . Demostrar que el algoritmo propuesto es correcto.

* C : cota superior \rightarrow existe un recorrido de s a t / $C(P) \leq c$

* Dijkstra nos devuelve el camino de peso mínimo de s a $t \rightarrow$ si el costo es $> c$
 \rightarrow \nexists otro camino ni recorrido de s a t que pese menos
 \rightarrow \nexists una arista / que cumpla con lo pedido

Algoritmo

1. Hago Dijkstra sobre G
2. Me fijo cuál es el peso del camino mínimo de s a t
3. Si el peso es $\leq c$ entonces recorro el camino buscando la arista de mayor peso y la devuelvo
4. Si el peso es $> c$ entonces aviso que no \exists una arista.

Dem

QVQ Algoritmo 1(G) devuelve una arista de peso máximo de entre aquellas que se encuentran en algún recorrido de s a t cuyo peso sea a lo sumo c .

* Si $\nexists P$ recorrido de s a t / $C(P) \leq c \rightarrow$ la arista que se nos pide \nexists

Como Algoritmo 1(G) hace 1^{er} Dijkstra y este es correcto vamos a tener el camino de peso mínimo de s a t . Como un camino es un recorrido es correcto haber usado Dijkstra. Además tampoco hay una restricción sobre el recorrido y solo se nos pide alguno, está bien haber usado Dijkstra. Como Dijkstra es correcto entonces si el peso del camino es mayor a c , \nexists otro camino y \therefore recorrido de menor peso \rightarrow es correcto decir que \nexists

la pista.

Si el peso del camino es a lo mismo C , como un camino es un recorrido, devolver la pista que \oplus pesa es correcto.