ALGORITMO de DIJKSTRA $\qquad \mathcal{V} = \{V_0, V_1, \dots, V_n\}$

Input: Grafo valorado con costes no negativos. Origen V_0. Destinos $V_1, \dots, V_n$

Inicialización: Metemos en un heap H los pares $(V_0, 0)$, (V_i, ∞) $i \in 1..n$.
Se trata de un heap de mínimos ordenado por la segunda componente y con acceso directo a cada elemento a través de la identidad de V_i.

Iteraciones: Sacamos del heap su cima (V_i, c_i) y hacemos $F(i) := c_i$, tras lo cual aplicamos relajación a cada arco (V_i, V_j), reduciendo c_j cuando cumpla que $c_j > F(i) + c(i,j)$.

Hechos: Para cada V_i accesible desde V_0 existe algún camino de coste mínimo entre todos ellos. Denotamos por m_i el coste del mismo, tomando $m_i = \infty$ para los vértices inaccesibles.

Notación: Denotamos por $\mathcal{V}_H$ los vértices todavía en el heap, y por $\mathcal{V}_{\bar{H}} = \mathcal{V} - \mathcal{V}_H$.

Invariantes: La secuencia de vértices que ha ido saliendo de $\mathcal{V}_H$, $V_{i_0}, \dots, V_{i_k}$ siempre cumple que: i) $\forall j \in 0..k \ \ F(i_j) = m_{i_j}$; ii) $m_{i_0} \le m_{i_1} \le \dots \le m_{i_k}$; iii) $\forall V_i \in \mathcal{V}_H \ \ m_{i_k} \le m_i$. Por otra parte, iv) $\forall V_i \in \mathcal{V}_H$ el valor c_i actual es el coste del camino más corto desde V_0 a V_i que antes de V_i sólo atraviesa vértices que ya están en $\mathcal{V}_{\bar{H}}$.

Comprobación: a) Al principio se cumplen trivialmente i), ii) y iii) por ser $\mathcal{V}_{\bar{H}} = \emptyset$, y también iv), porque para V_0 el camino vacío tiene coste 0, y en cambio no hay caminos desde V_0 a los demás V_i atravesando sólo vértices de $\mathcal{V}_{\bar{H}}$.

b) Veamos ahora que como consecuencia del invariante para el vértice V_i que ocupa la cima de H tenemos b1) $c_i = m_i$ y b2) $\forall V_j \in \mathcal{V}_H \ \ m_i \le m_j$.

- b2) Obviamente tenemos que $m_i \le c_i$. Si existiera $V_j \in \mathcal{V}_H$ con $m_i > m_j$ tendríamos $m_j \le c_i$ y el camino que genera m_j comienza en V_0, y por tanto en $\mathcal{V}_{\bar{H}}$. En algún momento ha de salir de $\mathcal{V}_{\bar{H}}$ pasando a algún $V_{j'} \in \mathcal{V}_H$ que verificaría $c_{j'} = m_{j'} \le m_j < c_i$, contradiciendo el que V_i ocupa la cima de $\mathcal{V}_H$.
- b1) Si no hubiera ningún camino más corto de V_0 a V_i que sólo atravesara vértices de $\mathcal{V}_{\bar{H}}$, repitiendo el razonamiento anterior volveríamos a encontrar un $V_{j'} \in \mathcal{V}_H$ con $c_{j'} < c_i$, lo que es imposible. Por tanto tendremos $c_i = m_i$.

c) En consecuencia, al pasar V_i a $\mathcal{V}_{\bar{H}}$ se mantienen i), ii) y iii)

d) Finalmente, al aplicar relajación sobre los arcos que salen de V_i recuperamos trivialmente iv).

Corolario: El algoritmo de Dijkstra calcula correctamente los costes de los caminos mínimos desde V_0 a cada vértice accesible V_i, y constata que los no accesibles lo son devolviendo para ellos $F(j) = \infty$. Además, el orden en que genera su salida es no decreciente con respecto al coste de los correspondientes caminos minimales. Por último, si al aplicar el relajamiento asociamos a cada vértice V_j un campo Π en el que recordamos la identidad del correspondiente origen V_i, la aplicación reiterada de Π como función sobre cada destino V_j para el cual $F(j) < \infty$ hasta llegar a V_0 nos produce un camino minimal revertido de V_0 a V_j.