

Algoritmo de Dijkstra

función Dijkstra(g: grafo; v: vértice) **devuelve** tabla[1..n] **de** natural

var

T: conjVert;
{en D se almacenan las distancias mínimas y en P el vértice predecesor}
D, P: tabla[1..n] **de** natural;
u, w: vértice;
val: natural;

fvar

inicio

T := ∅;
{inicializa las distancias de los vértices alcanzables desde el vértice origen}

para todo w **en** g **hacer**

D[w] := peso(g, v, w);

T := T ∪ {w}

fpara;

D[v] := 0;

T := T - {v};

repetir n-2 veces

{selección del mínimo w: $w \in T$ y $\forall u \in T: D[w] \leq D[u]$ }

val := ∞;

para todo u **en** T **hacer**

si D[u] ≤ val **entonces**

w := u;

val := D[u]

fsi

fpara;

{se marca w como vértice tratado}

T := T - {w};

{se recalculan las nuevas distancias mínimas}

para todo u **en** T **hacer**

si D[w] + peso(g, w, u) < D[u] **entonces**

D[u] := D[w] + peso(g, w, u);

P[u] := w;

fsi

fpara

frepetir

devuelve D

ffunción