

1 Algorytm Dijkstry

Start

Wczytaj graf skierowany *graph* w postaci macierzowej, *source* oraz *destination*

vertex_count \leftarrow ilość wierzchołków *graph*

weights \leftarrow tablica długości *vertex_count*

history \leftarrow tablica długości *vertex_count*

Dla $i \leftarrow 0$ **do** *vertex_count* **wykonuj**

 | *weights*[*i*] $\leftarrow \infty$

 | *history*[*i*] \leftarrow brak

koniec

pending_vertices \leftarrow set pojemności *vertex_count*

Dla $i \leftarrow 0$ **do** *vertex_count* **wykonuj**

 | Dodaj *i* do *pending_vertices*

koniec

weights[*source*] $\leftarrow 0$

Gdy długość *pending_vertices* > 0 **wykonuj**

 | *vertex* \leftarrow element z *pending_vertices* o najmniejszej wartości

Jeżeli *vertex* = *destination* **to**

 | Wyjdź z pętli

koniec

weights[*vertex*]

 Usuń wartość *vertex* z *pending_vertices*

Dla *neighbor* $\leftarrow 0$ **do** *vertex_count* **wykonuj**

 | *distance* \leftarrow *graph*[*vertex*][*neighbor*]

Jeżeli *distance* ≤ 0 **to**

 | Przejdź do kolejnej iteracji pętli

koniec

calculated_weight \leftarrow *weights*[*vertex*] + *distance*

Jeżeli *calculated_weight* $<$ *weights*[*neighbor*] **to**

 | *weights*[*neighbor*] \leftarrow *calculated_weight*

 | *history*[*neighbor*] \leftarrow *vertex*

koniec

koniec

koniec

path \leftarrow pusta lista

vertex \leftarrow *destination*

Gdy *vertex* $\neq -1$ **wykonuj**

 | Wstaw wartość *vertex* na pozycji 0 do *path*

 | *vertex* \leftarrow *history*[*vertex*]

koniec

Wypisz ścieżkę *path* oraz sumaryczny dystans *weights*[*destination*]

Stop