

# 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 *weights*[*vertex*]

    | Usuń wartość *vertex* z *pending\_vertices*

    | **Dla** *neighbor*  $\leftarrow 0$  **do** *vertex\_count* **wykonuj**

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

        | **Jeżeli** *distance*  $\leq 0$  **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