

## 1 Algorytm A\*

Start

Wczytaj graf skierowany *graph* w postaci macierzowej, *source*, *destination* oraz funkcję heurystyczną przewidywanej odeległości *approximate\_distance*

*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*]  $\leftarrow$  **para**  $\infty$ ,  $\infty$

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

**koniec**

*pending\_vertices*  $\leftarrow$  set pojemności *vertex\_count*

Dodaj *source* do *pending\_vertices*

*weights*[*source*]  $\leftarrow$  **para** 0, *approximate\_distance*(*source*)

**Gdy** długość *pending\_vertices*  $> 0$  **wykonuj**

    | *vertex*  $\leftarrow$  element z *pending\_vertices* o najmniejszej wartości *weights*[*vertex*][1]

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

        | **Wyjdź z pętli**

**koniec**

    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*][0] + *distance*

**Jeżeli** *calculated\_weight*  $<$  *weights*[*neighbor*][0] **to**

            | *weights*[*neighbor*]  $\leftarrow$  **para** *calculated\_weight*,

            | *calculated\_weight* + *approximate\_distance*(*neighbor*)

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

            Dodaj *neighbor* do *pending\_vertices*

**koniec**

**koniec**

**koniec**

**Jeżeli** długość *pending\_vertices*  $< 1$  **to**

    | Wypisz komunikat o braku rozwiązań.

    | **Stop**

**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*][0]

**Stop**