

1 Algorytm A*

Start

Wczytaj graf skierowany $graph$ w postaci macierzowej, $source$, $destination$ oraz funkcję heurystyczną przewidywanej odległości $approximate_distance$

$vertex_count \leftarrow$ 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 ∞, ∞

| $history[i] \leftarrow$ 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

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