

## Algorytmy grafowe

### Opis implementacji

Plansza zostaje przekształcona w obiekt klasy *Graph* poprzez wczytanie wszystkich wierszy jako elementów klasy *Dict* (która jest klasą bazową naszej klasy *Graph*). Kluczami w powstałym słowniku (lub mapie, zależnie od nazewnictwa) jest para koordynatów (x, y), a wartościami są koszty wejścia na dane pola.

Do znalezienia najkrótszej ścieżki jest wykorzystywany standardowy algorytm Dijkstry.

Najpierw tworzymy tablicę *costs* odległości od startu dla wszystkich zbadanych wierzchołków grafu. Następnie tworzymy kolejkę priorytetową *queue* wszystkich wierzchołków grafu wraz z najkrótszą znaną ścieżką od startu do danego punktu. Priorytetem tej kolejki jest odległość od startu do danego punktu.

Dopóki kolejka nie jest pusta, pobieramy jej następny element. Jeżeli znamy już krótszą trasę do tego pola, ignorujemy dany element. Jeżeli pobrany element to pole końcowe, zwracamy tą ścieżkę.

Jeżeli nic z powyższych nie zostało spełnione, znajdujemy wszystkie pola sąsiadujące z aktualną pozycją wraz z kosztem wejścia na nie.

Jeżeli nowy koszt drogi do sąsiada jest niższy od już znanego, zapisujemy go do tablicy kosztów oraz wypychamy do kolejki *queue* element wraz z nową trasą.

Jeżeli elementy z kolejki się skończyły, a algorytm dalej nie znalazł trasy, to rzucaamy wyjątek *NoPathExists*. W poprawnym grafie sytuacja ta nie zachodzi.

### Wyniki

Plansza 1 [*input.txt*] (z treści zadania) i jej rozwiązanie:

111122	*****
104122	*0****
941111	*41***
996411	**6***
990411	**0***
991111	*****

Wynik planszy nr. 1 różni się od zadanego w treści zadania, ponieważ istnieją na niej dwie ścieżki o tym samym koszcie. W naszym przypadku algorytm wybrał tą drugą. Aby wymusić użycie pierwszej ścieżki, w planszy 2 zastosowano zwiększone koszty w odpowiednich miejscach planszy 1.

Plansza 2 [*input1.txt*] i jej rozwiązanie:

111122	*111**
105122	*0*1**
951111	***11*
996411	****1*
990411	**0*1*
991111	**111*

Plansza 3 [input2.txt] i jej rozwiązanie:

123429567892	*****
105125932192	*0512*****
956789134522	****8*****
996432198722	****321*****
997445678992	*****6*****
991876543212	*****5432**
111192165122	*****1**
145126532122	*****1**
951642452222	*****22*
996454224522	*****2*
995423414302	*****0*
991875875622	*****

Plansza 4 [input3.txt] i jej rozwiązanie:

824711891596	*****
307421428661	*0742*****
483547722764	****4*****
774914379571	****1*****
429231667489	****3*****
853937512116	****3*****
562316376466	****16*****
841884647895	****4*****
966985883769	****5*****
399797189295	****71*****
551976334202	****33420*
313478993166	*****