

Jednoosobowa gra logiczna

Małgorzata Śleszyńska-Nowak

Gra odbywa się na prostokątnej planszy. Pionek początkowo znajduje się na pozycji startowej (0,0). Gracz ma dostępną listę dozwolonych ruchów. Ruchy są postaci (di, dj) , gdzie di wskazuje o ile wierszy należy przesunąć pionek, a dj o ile kolumn (są to wartości całkowite, nieujemne). Każdy ruch ma przypisany swój koszt.

Celem **etapu 1** jest doprowadzenie pionka do ostatniego wiersza planszy (pozycję w ostatnim wierszu i dowolnej kolumnie będziemy nazywać pozycją docelową) przy możliwie najmniejszym koszcie.

W **etapie 2** mamy dodatkowe założenie - każdy z dozwolonych ruchów może zostać wykonany co najwyżej raz.

Dane

- $int\ n$ – liczba całkowita dodatnia, wysokość planszy (liczba wierszy),
- $int\ m$ – liczba całkowita dodatnia, szerokość planszy (liczba kolumn),
- $((int\ di, int\ dj)\ step, int\ cost)[\]\ moves$ – tablica dozwolonych ruchów, gdzie:
 - $step.di, step.dj$ – liczby całkowite nieujemne, oznaczają odpowiednio przesunięcie w pionie (zmiana wiersza) i w poziomie (zmiana kolumny),
 - $cost$ – liczba całkowita dodatnia, koszt danego ruchu.

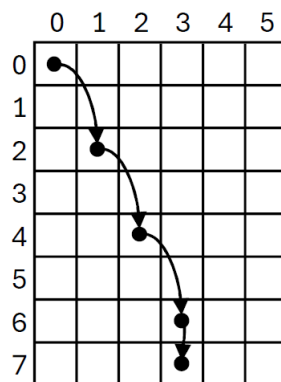
Szukane

W obu wariantach należy zwrócić: $(bool\ result, int\ cost, (int\ i, int\ j)[\]\ path)$, gdzie:

- $bool\ result$ – $true$ jeżeli możliwe jest osiągnięcie pozycji docelowej, $false$ wpp.,
- $cost$ – minimalny koszt trasy pionka startującego z pozycji startowej i kończącego w pozycji docelowej,
- $(int\ i, int\ j)[\]\ path$ – tablica kolejnych pozycji pionka (włącznie z pozycją startową i końcową), dla każdej pozycji (i, j) to krotka (indeks wiersza, indeks kolumny).

Przykład

Plansza 8 x 6, $moves = [((1, 1), 4), ((2, 2), 6), ((2, 1), 5), ((1, 0), 3), ((0, 1), 3)]$.



Rysunek 1: Przykładowa rozgrywka. Etap 1.

Rozwiązanie etapu 1:

- trasa istnieje,
- minimalny koszt: 18,
- trasa o minimalnym koszcie: $(0, 0) \rightarrow (2, 1) \rightarrow (4, 2) \rightarrow (6, 3) \rightarrow (7, 3)$.
Wykonujemy po kolei ruchy: $((2, 1), 5), ((2, 1), 5), ((2, 1), 5), ((1, 0), 3)$.

Rozwiązanie etapu 2: trasa nie istnieje, nie można osiągnąć pozycji docelowej wykonując każdy ruch co najwyżej raz.

Punktacja

- Etap pierwszy (**1.5p**) – minimalny koszt trasy (**0.5p**), trasa realizująca ten koszt (**1p**).
- Etap drugi (**1p**)

Uwagi i wskazówki

- Gdy istnieje kilka możliwych tras realizujących minimalny koszt, można zwrócić dowolną z nich.
- Wymagana złożoność czasowa dla obu etapów $O(knm)$, gdzie k to liczba dozwolonych ruchów, skorzystaj z programowania dynamicznego.