**Първо домашно - Търсене на път до цел**

В задачата ще помогнем на чудовището Митко да се ориентира в лабиринт. То е много гладно и, за да пести енергия, е изключило способността си да минава през стени. Сега трябва да намери най-лекия път до храната си. Ето един примерен лабиринт:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** |
| **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  | N | N | N |  |  |  |
| **3** |  |  | N | N |  |  | N | N |  | N |  |  |  |
| **4** |  | ~ |  | N |  |  | ~ |  |  | N | N | N |  |
| **5** |  |  |  | N |  |  | ~ | X |  | N |  |  |  |
| **6** | M | ~ |  | N | N | N | N | N | N | N |  |  |  |
| **7** |  |  |  | N |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |  | ~ | ~ | ~ |  |

Означения в лабиринта:

**Празна плочка** – свободен коридор

**N** – стена в лабиринта

**~** - наводнен терен

**М** – Митко, гладно чудовище, което има нужда от помощ

**X** - храна, целта на Митко

Правила за движение:

1. Митко се движи свободно във всички посоки, включително диагоналите
2. Когато е гладен Митко не може да минава през стени. (а той е гладен)
3. Цената на всяко движение на север (N), изток (E), запад (W) или юг (S) е 1.
4. Цената на всяко движение по диагонал (NE, SE, SW, NW) е 1.5.
5. Митко обожава вода. Ако пътят му минава през водна площ, той задължително ще остане да поплува. Цената на всяко движение от водна площ без значение на къде е 2.

И така:

**Да се напише програма, която при задаване на лабиринт (матрица от символи) намира оптималния път от начални координати до крайни координати.**

**При стартиране на вашата програма тя ще получава 5 аргумента – пълен път до файл и координати на две точки в цели числа. Файлът ще бъде csv и ще съдържа матрицата на лабиринта. Първата точка е началната позиция на Митко, а втората – къде се намира храната.**

**Програмата ви трябва да извежда на конзолата списък от последователни точки, през които Митко минава по пътя си до храната.**

Ето пример как може да изглежда основната ви функция (Java, но както знаете – езикът е без значение):

public static void main (String[] args) {

Maze maze = new Maze(args[0]);

Point start = new Point(args[1], args[2]));

Point end = new Point(args[3], args[4]));

List<Point> path = maze.findPath(start, end);

for (Point step : path) {

System.out.print(step.toString());

if (!step.equals(end)) {

System.out.print(" ,");

}

}

}

**Програмата трябва да намира оптимално решение (пътя с най-ниска цена) и да го извежда в списък от последователните стъпки – координати в лабиринта:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** |
| **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** |  |  | \* | \* |  |  |  | N | N | N |  |  |  |
| **3** |  | \* | N | N | \* |  | N | N |  | N |  |  |  |
| **4** | \* | ~ |  | N |  | \* | ~ |  |  | N | N | N |  |
| **5** | \* |  |  | N |  |  | \* | X |  | N |  |  |  |
| **6** | M | ~ |  | N | N | N | N | N | N | N |  |  |  |
| **7** |  |  |  | N |  |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |  | ~ | ~ | ~ |  |

[(А, 5), (А, 4), (B, 3), (C,2) (D, 2), (E, 3), (F, 4), (G, 5) (H, 5)]

**Примерни елементи на решението направихме на дъската с по време на упражнения за пътната карта и намирането на път между два града.**