# A* algorithm. Part 1

Cost: 5 | Solved: 52

**Memory limit: 256** MBs

**Time limit:** 1 s

**Input**: input.txt

**Output:** output.txt

**Task:**

You are given a maze **-** an **_n_** * **_m_** matrix.

You have to find the shortest route between two vertices, implementing **_A* (A star) algorithm_**.

_Take the Manhattan distance as your heuristic._

**Input:**

The first line contains two integers **_n_** and **_m_** - the size of the maze *(1 <= **n, m** <= 1000)*.

The second line contains four numbers: **_x₁_**, **_y₁_** – the coordinates of the initial vertex, **_x₂_**, **_y₂_** – the coordinates of the final vertex.

The next **_n_** lines should contain the maze itself, where 0 means open space and **-1** is an obstruction.

_The coordinate of the top left corner of the maze is (1; 1), of the bottom right – (**m**; **n**)._

**Output:**

The first line should contain the number of transitions required to reach the final vertex.

Next lines should contain the transitions themselves, which means that you consecutively output the coordinates of the vertexes of your route.

_Every vertex should be separated with a line feed._

**Example:**

| Input | Output |
|---|---|
| 4 5<br><br>1 1 5 4<br><br>0 -1 0 0 0<br><br>0 -1 0 -1 0<br><br>0 0 0 -1 0<br><br>0 0 0 -1 0 | 1 2<br><br>1 1<br><br>1 2<br><br>1 3<br><br>2 3<br><br>3 3<br><br>3 2<br><br>3 1<br><br>4 1<br><br>5 1<br><br>5 2<br><br>5 3<br><br>5 4 |