Let’s help Mitko one more time

Mitko is hungry again and this time the labyrinth is more complicated than before. So he needs your help again.

Now as you know when Mitko is hungry he can not go through walls in order to save the energy. He needs to find the most optimal path to his food. Here is an example labyrinth:

	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										~	~	~	

What this means:
Empty tile – Mitko can just pass
N – Wall in the labyrinth, Mitko cannot pass
~ - The area is flooded, Mitko can pass, but it will take time
M – This is Mitko
X – The food, your target

Possible moves and cost:

1. Mitko can go free in all directions – north (up), south(down), east(right), west(left). The cost for each move is 1.
2. Mitko can go via the diagonals (NE, SE, SW, NW), but this costs more energy. The cost for moving in diagonal is 1.5.
3. Mitko can go via the flooded, and he is really fast swimmer, too. However, Mitko loves swimming. So every time he goes via a water tile, Mitko will stay to play. Each move away from a water tile (straight or diagonal) is at cost 2.

Make a program that reads a matrix of symbols, that represent a labyrinth and finds the optimal path from a given starting position to a given target position.

The program will get a path to a file (csv format) that contains the labyrinth. You will get the position of Mitko (x and y) and the position of the food (x and y)

At the end you should have a list of steps, that Mitko should go through to get to the food.

Find the optimal path to lead Mitko to his food by spending the least amount of energy.

	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										~	~	~	

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