

For the problem above, you are given a python program that implements *Breath First Search (BFS)* and *Uniform Cost Search (UCS)* search strategies. You are required to analyze BFS in order to get familiar with programming in python. Then you need to debug the UCS function implemented in the program in order for it to function correctly.

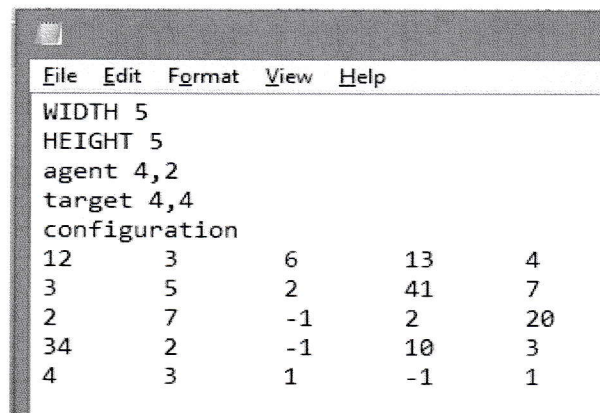
Files:

searchPath.py: python program to solve the navigation problem

inputfilex.txt: input files to the python program describing the grid environment. 'x' indicates the input file number

Input:

6 input files are provided, which are named from "inputfile0.txt" to "inputfile5.txt". The format of a sample input file is given below.



```
File Edit Format View Help
WIDTH 5
HEIGHT 5
agent 4,2
target 4,4
configuration
12      3      6      13      4
3       5      2      41      7
2       7      -1     2      20
34      2      -1     10     3
4       3      1      -1     1
```

The size of the input grid is defined by "WIDTH" and "HEIGHT" which corresponds to the number of cells in each row and the number of cells in each column respectively. Cell(0,0) is top-left cell of the input grid. The (x,y) coordinates of the agent and target are given in the third and fourth line respectively, with x presents the vertical position, y presents the horizontal position. The blocked cells are shown as "-1". The unblocked cells are shown as positive numbers which also indicate the costs to traverse them. The numbers in the same line are separated by tab.