Output:

The program shows the path between the agent cell and the target cell. Blocked cells are displayed as "X", the Agent cell as "A", the target cell as "T" and the cells on the path are displayed by ".". Other cells are displayed as single white space. In the end the program displays the cost of traversing from agent cell to target cell through the path determined by the search strategy selected. The following figure shows an example of the output:

Running the program:

To run the program, use the following command:
python searchPath.py –i *inputfile_name* –s *search_strategy*
where,
inputfile_name is the name of the input file.
search_strategy is the selected search strategy (1=BFS and 2=UCS)

The following command shows an example where it selects "inputfile0.txt" as the input file and BFS as the search strategy:
python searchPath.py –i inputfile0.txt –s 1

NOTES:
1) You can download the python program and input files from the course website (https://courses.uscden.net/d2l/home/7593). Under the "Content" tab find "Homework". Under that you will find all the files in "HW1" folder.
2) We use Python 2.7 for this implementation. (https://www.python.org/download/releases/2.7/)
3) The program assumes that the input files are in the same folder as the program.

Answer the following questions:

a) [10 Points] Debug the UCS function in the program to make it function as described in Figure 3.14 in the book. Indicate the modification/addition to the program in terms of python code. Use the line number in the program as the reference to where you would modify/add code. Explain why each modification/addition was required.

b) [6 Points] For each input file, show the path from Agent cell to target cell and report the path cost for the debugged UCS function.

A* can be thought as an enhanced version of UCS that makes use of additional knowledge for estimating the cost of a path heuristics. If we were to convert the UCS program to do A* search, we would need to find heuristic for estimating the cost in this grid environment. With that in mind:

c) [2 Points] Give a simple argument that shows that the heuristic function directly using Manhattan Distance is consistent.

d) [2 Points] Design one admissible heuristic function other than using Manhattan Distance.