CENG 462

Introduction to Artificial Intelligence

Fall '2017-2018

Take Home Exam 1

Due date: 12 November 2017, Sunday, 23:59

SLIDING BLOCKS PUZZLE

The puzzle in this exam is the sliding blocks puzzle. It is a simple puzzle where it starts in a certain state and you move the blocks one unit at a time, namely *right*, *left*, *up* or *down*, to the empty spaces in certain combinations to get a certain block out of the grid, i.e. the final state. Figure 1 shows the initial state of the game, by moving blocks into the possible empty spaces the desired final state is reached (given last). Note that there can be many more final states but for the sake of simplicity you will be given only a subset of them.

You will be given a number of tasks to solve with *A*-search* algorithm. You will use the *Manhattan distance* heuristic and a heuristic of your own. You need to prepare a report to give a proof that your heuristic is admissible.

Your program will start reading the input file, *hw1.inp*, given. Then it should wait \n (enter) character to solve the first task. After, it should output only the selected paths by the algorithm (like in the figure) in matrix form from start state till final state to *stdout*. After, it should wait an \n character to solve the second task, and so on.

An example input file, *the1.inp*, has the following structure :

2
0
4 3 3 1
S
0 0 0
1 2 0
0 3 3

```
0 3 3
F
3 3 0
3 3 0
2 0 0
1 0 0
1
4 3 2 2
S
0 0 0
0 1 0
0 2 0
0 2 0
F
0 2 0
1 2 0
0 0 0
0 0 0
F
0 2 0
0 2 1
0 0 0
0 0 0
```

The first line of the file states the number of independent tasks you are dealing with. The first line of the following lines is either 0 or 1. For 0 you should use *Manhattan distance* whereas for 1 you should use *your heuristic*. Then the next line indicates the number of *rows, columns*, *pieces* and the given final states in the puzzle. Beneath, there is a character indicating the *S*tart state, then the following lines defines the grid of the start state. Each new number in the grid defines a new block in the puzzle. After there is a character indicating the *F*inal state, then the following lines will indicate the grid of the final state. This process repeats for all of the tasks.
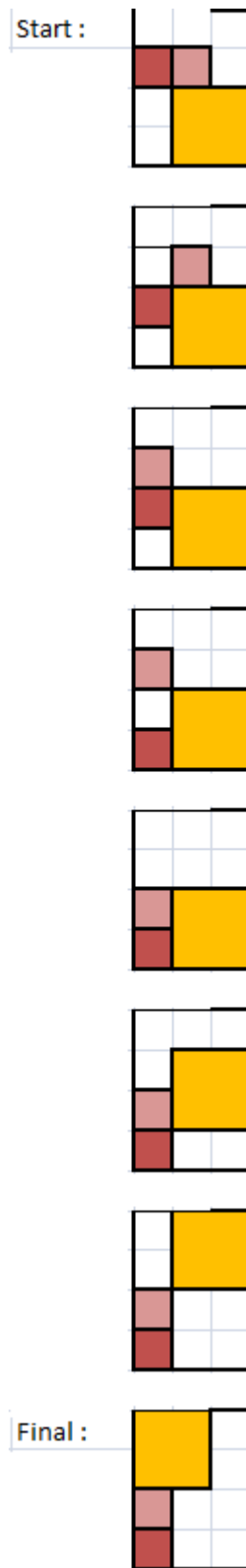
Figure 1: An example play

1   Regulations

1. Your code should be in Python.

2. **Late Submission: Not Allowed**

3. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.

4. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.

5. **Evaluation:** The .py file will be checked for plagiarism automatically using "black-box" technique and manually by assistants, so make sure to obey the specifications.

2   Submission

Submission will be done via COW. Zip your code file named "the1.py" and your report "the1.pdf" into archive named "the1.rar". Do not put a folder inside the archive.

Note: You cannot submit any other files. Don't forget to make sure your .py file is successfully compiled in Inek machines.