Middle East Technical University Department of Computer Engineering

CENG 462

Artificial Intelligence

Fall '2019-2020

Homework 1

Due date: 29 October 2019, Tuesday, 23:55

1 Objectives

This assignment aims to assist you to expand your knowledge on informed search in the cases of A* and Iterative Deepening A* algorithms.

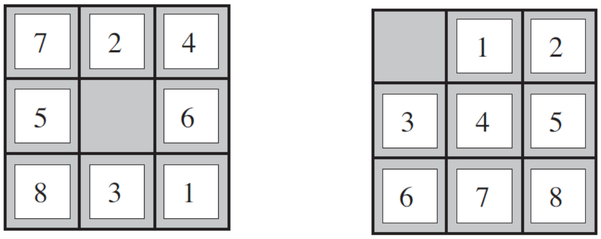
2 Problem Definition

In this assignment, you are going to solve a given N-puzzle by using A* and Iterative Deepening A* algorithm.

You already know what N-puzzle is, but just to refresh your memory, here is the definition of N-puzzle:

N-puzzle is a game consisting of a k x k board with $N = k^2 - 1$ sliding tiles and a blank space where the aim is to reach a specific configuration from a given configuration by moving the tiles near the blank space.

Here is an example of initial and goal configurations for 8-puzzle (where k = 3) from the textbook:



(a) initial state (b) goal state

Figure 1: Example configurations as initial and goal states for 8-puzzle (taken from [1])

3 Specifications

- You are going to implement A* and Iterative Deepening A* algorithms (you can refer the lecture slides for pseudocodes.) in python.
- You will use the **total Manhattan distance** as the heuristic function. For instance, it is 18 for the example given in Section 2.
- The task will be given from standard input and the result will be printed to standard output.
- Input will consist of:
  - the method you are going to use for the given task ("A*" or "IDA*" for Iterative Deepening A*)
  - M as the maximum total estimated cost allowed in the method
  - k defining the dimension of board,
  - the initial configuration of the board,
  - the goal configuration of the board.
- Output will consist of "SUCCESS" or "FAILURE" stating whether the goal is reached or not respectively. If the solution exists, all configurations on the path from the initial configuration to the goal configuration will be printed as well.
- While expanding a node you are required to try to move **up**, **down**, **left**, **right** in that order (Moving up means sliding the tile below the blank upward, similar for other directions.)
- When searching for the state with the minimum cost you will select the **first** one for tie-breaking.

IMPORTANT NOTE: Last two items are crucial for the black-box evaluation. Please avoid any violation of them, in order not to lose any points redundantly.

4 Sample I/O

Input 1:

```
A*
50
3
3 2 5
6 4 1
7 _ 8
_ 1 2
3 4 5
6 7 8
```

Output 1:

```
SUCCESS

3 2 5
6 4 1
7 _ 8

3 2 5
6 4 1
_ 7 8

3 2 5
_ 4 1
6 7 8

3 2 5
4 _ 1
6 7 8

3 2 5
4 1 _
6 7 8

3 2 _
4 1 5
6 7 8

3 _ 2
4 1 5
6 7 8

3 1 2
4 _ 5
6 7 8

3 1 2
_ 4 5
6 7 8

_ 1 2
3 4 5
6 7 8
```

Input 2:

```
IDA*
50
3
3 2 5
6 4 1
7 _ 8
_ 1 2
3 4 5
6 7 8
```

Output 2:

```
SUCCESS

3 2 5
6 4 1
7 _ 8

3 2 5
6 _ 1
7 4 8

3 2 5
6 1 _
7 4 8

3 2 _
6 1 5
7 4 8

3 _ 2
6 1 5
7 4 8

3 1 2
6 _ 5
7 4 8

3 1 2
6 4 5
7 _ 8

3 1 2
6 4 5
_ 7 8

3 1 2
_ 4 5
6 7 8

_ 1 2
3 4 5
6 7 8
```

Input 3:

```
A*
20
4
2 1 3 4
5 6 7 8
9 10 11 12
13 14 15 _
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 _
```

Output 3:

```
FAILURE
```

Input 4:

```
IDA*
20
3
1 2 3
4 5 6
7 8 _
1 2 3
4 5 6
8 7 _
```

Output 4:

```
FAILURE
```

5 Regulations

1. **Programming Language:** You must code your program in Python. Your submission will be tested in inek machines. So make sure that it can be executed on them.

2. **Implementation:** You have to code your program by only using the functions in standard module of python. Namely, you **cannot** import any module in your program. The only exception for this rule is the `copy` module which may be useful with the `deepcopy` function to cope with aliasing list problem in python.

3. **Late Submission:** No late submission is allowed.

4. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating (any kind of code sharing and codes taken from internet included) will be punished according to the university regulations.

5. **Discussion:** You must follow the OdtuClass for discussions and possible updates on a daily basis.

6. **Evaluation:** Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications. A reasonable timeout will be applied according to the complexity of test cases in order to avoid infinite loops due to an erroneous code.

6 Submission

Submission will be done via OdtuClass system. You should upload a **single** python file named in the format **<your-student-id>_hw1.py** (i.e. e1234567_hw1.py).

References

[1] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach.* Malaysia; Pearson Education Limited,, 2016.