



CS-461

Artificial Intelligence

Homework #2

Group Name: Enigma

Group Members:

1. Ege Yosunkaya - 21402025 / CS
2. Kerem Ayöz - 21501569 / CS
3. Erkut Alakuş - 21101413 / CS

We run our program multiple times and results of them are in below:

1

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(2, 2, 1, 0, 'Right')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(3, 1, 0, 1, 'Right')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(3, 0, 0, 2, 'Left')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(0, 2, 3, 0, 'Right')  
(1, 2, 2, 0, 'Right')  
(1, 0, 2, 2, 'Right')  
(2, 0, 1, 2, 'Left')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

2

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(2, 2, 1, 0, 'Right')  
(3, 1, 0, 1, 'Right')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(1, 2, 2, 0, 'Right')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(3, 0, 0, 2, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(0, 2, 3, 0, 'Right')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(2, 0, 1, 2, 'Left')  
(0, 2, 3, 0, 'Left')  
.....
```

3

Path to the goal

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

Closed list

.....
(3, 2, 0, 0, 'Left')
(2, 2, 1, 0, 'Right')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(1, 2, 2, 0, 'Right')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(0, 2, 3, 0, 'Right')
(1, 0, 2, 2, 'Right')
(3, 0, 0, 2, 'Left')
(2, 0, 1, 2, 'Left')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

4

Path to the goal

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

Closed list

.....
(3, 2, 0, 0, 'Left')
(2, 2, 1, 0, 'Right')
(3, 1, 0, 1, 'Right')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(1, 2, 2, 0, 'Right')
(2, 0, 1, 2, 'Right')
(3, 0, 0, 2, 'Left')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(0, 2, 3, 0, 'Right')
(1, 1, 2, 1, 'Left')
(2, 0, 1, 2, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

5

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 1, 0, 1, 'Right')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Right')  
(2, 2, 1, 0, 'Left')  
(1, 2, 2, 0, 'Right')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(3, 0, 0, 2, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

6

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 1, 0, 1, 'Right')  
(3, 0, 0, 2, 'Right')  
(2, 2, 1, 0, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(1, 2, 2, 0, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(3, 0, 0, 2, 'Left')  
(1, 2, 2, 0, 'Left')  
(0, 2, 3, 0, 'Right')  
(1, 0, 2, 2, 'Right')  
(2, 0, 1, 2, 'Left')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

7

Path to the goal

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

Closed list

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(3, 1, 0, 1, 'Right')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(1, 2, 2, 0, 'Right')
(2, 2, 1, 0, 'Right')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(3, 0, 0, 2, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(0, 2, 3, 0, 'Right')
(1, 0, 2, 2, 'Right')
(2, 0, 1, 2, 'Left')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

8

Path to the goal

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Left')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(1, 1, 2, 1, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

Closed list

.....
(3, 2, 0, 0, 'Left')
(3, 0, 0, 2, 'Right')
(3, 1, 0, 1, 'Left')
(3, 1, 0, 1, 'Right')
(2, 1, 1, 1, 'Right')
(2, 2, 1, 0, 'Right')
(2, 2, 1, 0, 'Left')
(1, 2, 2, 0, 'Right')
(2, 0, 1, 2, 'Right')
(2, 1, 1, 1, 'Left')
(3, 0, 0, 2, 'Left')
(1, 1, 2, 1, 'Right')
(1, 2, 2, 0, 'Left')
(1, 0, 2, 2, 'Right')
(0, 2, 3, 0, 'Right')
(1, 1, 2, 1, 'Left')
(2, 0, 1, 2, 'Left')
(0, 1, 3, 1, 'Right')
(0, 2, 3, 0, 'Left')
.....

9

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(2, 2, 1, 0, 'Right')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(1, 2, 2, 0, 'Right')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(3, 0, 0, 2, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(0, 2, 3, 0, 'Right')  
(1, 0, 2, 2, 'Right')  
(2, 0, 1, 2, 'Left')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

10

Path to the goal

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

Closed list

```
.....  
(3, 2, 0, 0, 'Left')  
(3, 0, 0, 2, 'Right')  
(2, 2, 1, 0, 'Right')  
(3, 1, 0, 1, 'Right')  
(3, 1, 0, 1, 'Left')  
(2, 1, 1, 1, 'Right')  
(2, 2, 1, 0, 'Left')  
(1, 2, 2, 0, 'Right')  
(2, 0, 1, 2, 'Right')  
(2, 1, 1, 1, 'Left')  
(3, 0, 0, 2, 'Left')  
(1, 1, 2, 1, 'Right')  
(1, 2, 2, 0, 'Left')  
(1, 0, 2, 2, 'Right')  
(1, 1, 2, 1, 'Left')  
(0, 1, 3, 1, 'Right')  
(0, 2, 3, 0, 'Left')  
.....
```

As we can see, different runs may give different closed lists because of the randomness of the non-deterministic search. However, whatever the tried states are, we get only one solution from our program. This implies that this puzzle has a unique solution.

Our program adds the visited node to the closed list and expanding that visited node to find new paths. After finding new paths, it adds them into the queue in random positions. Therefore it has the mechanism of the non-deterministic search.

For instance after the first expansion of the initial state, program expands randomly one of the newly created states by getting a path's lastly visited state that is in the head of the queue. Therefore, we get different results in each of this run.

If all these random insertions are done to the 1st position in the queue, non-deterministic search becomes depth-first search and program can complete finding solution in 13 steps. Also if all these random insertions are done to the last position in the queue, non-deterministic search becomes breadth-first search and program can complete finding solution in 18 steps. Because of that low probabilities, our program generally complete finding path in 16-18 steps, which is between or equal to these results as expected.

As a consequence, our program finds the unique solution of that problem by trying different solutions. Non-deterministic search gives the expected results, it may behave like depth-first search or breadth-first search. The randomness of the insertions to queue determine its behaviour.