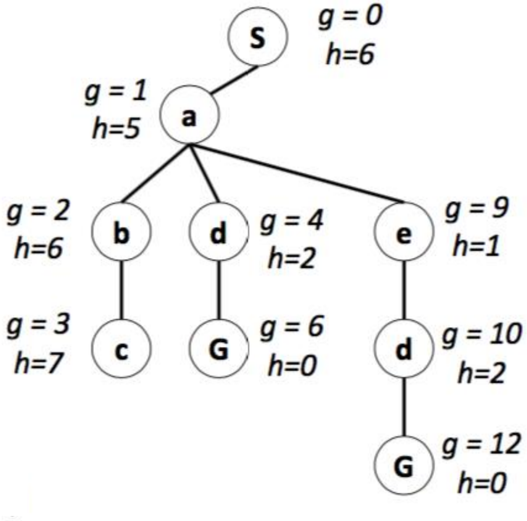
Artificial Intelligence
Assignment 1

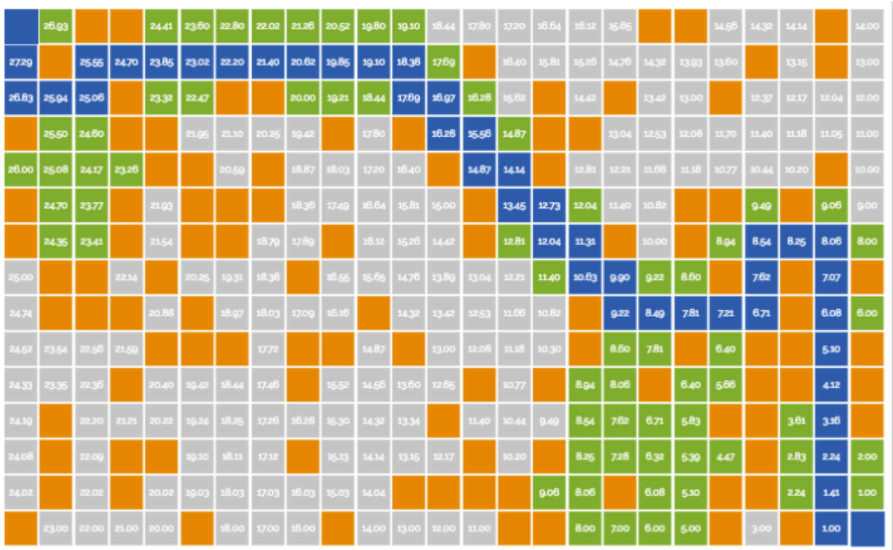
CSI4108-02
Spring, 2018

1. Introduction

- **We will find optimal solution for given mazes using A* search algorithm.**



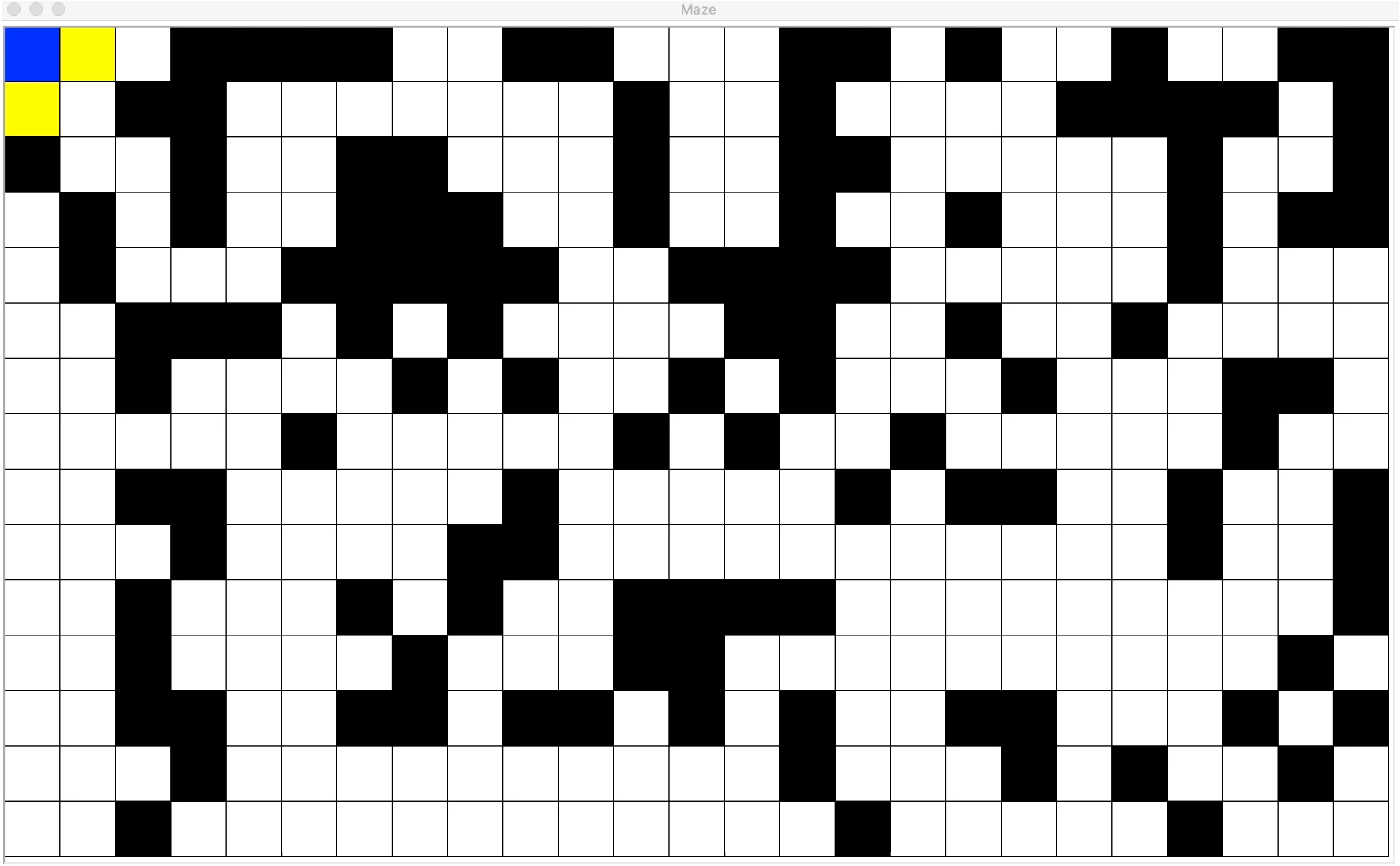
A* search algorithm



	26.93			24.41	23.60	22.80	22.02	21.26	20.52	19.80	19.10	18.44	17.80	17.20	16.64	16.12	15.65			14.56	14.32	14.14		14.00
27.29		25.55	24.70	23.85	23.02	22.20	21.40	20.62	19.85	19.10	18.38	17.69		16.40	15.81	15.26	14.76	14.32	13.93	13.60		13.15		13.00
26.83	25.94	25.06		23.32	22.47			20.00	19.21	18.44	17.69	16.97	16.28	15.62		14.42		13.42	13.00		12.37	12.17	12.04	12.00
	25.50	24.60			21.95	21.10	20.25	19.42		17.80		16.28	15.56	14.87			13.04	12.53	12.08	11.70	11.40	11.18	11.05	11.00
26.00	25.08	24.17	23.26			20.59		18.87	18.03	17.20	16.40		14.87	14.14		12.81	12.21	11.66	11.18	10.77	10.44	10.20		10.00
	24.70	23.77		21.93				18.36	17.49	16.64	15.81	15.00		13.45	12.73	12.04	11.40	10.82			9.49		9.06	9.00
	24.35	23.41		21.54			18.79	17.89		16.12	15.26	14.42		12.81	12.04	11.31		10.00		8.94	8.54	8.25	8.06	8.00
25.00			22.14		20.25	19.31	18.38		16.55	15.65	14.76	13.89	13.04	12.21	11.40	10.63	9.90	9.22	8.60		7.62		7.07	
24.74				20.88		18.97	18.03	17.09	16.16		14.32	13.42	12.53	11.66	10.82		9.22	8.49	7.81	7.21	6.71		6.08	6.00
24.52	23.54	22.56	21.59				17.72			14.87		13.00	12.08	11.18	10.30		8.60	7.81		6.40			5.10	
24.33	23.35	22.36		20.40	19.42	18.44	17.46		15.52	14.56	13.60	12.65		10.77		8.94	8.06		6.40	5.66			4.12	
24.19		22.20	21.21	20.22	19.24	18.25	17.26	16.28	15.30	14.32	13.34		11.40	10.44	9.49	8.54	7.62	6.71	5.83			3.61	3.16	
24.08		22.09			19.10	18.11	17.12		15.13	14.14	13.15	12.17		10.20		8.25	7.28	6.32	5.39	4.47		2.83	2.24	2.00
24.02		22.02		20.02	19.03	18.03	17.03	16.03	15.03	14.04					9.06	8.06		6.08	5.10			2.24	1.41	1.00
	23.00	22.00	21.00	20.00		18.00	17.00	16.00		14.00	13.00	12.00	11.00			8.00	7.00	6.00	5.00		3.00		1.00	

Maze

2. Example

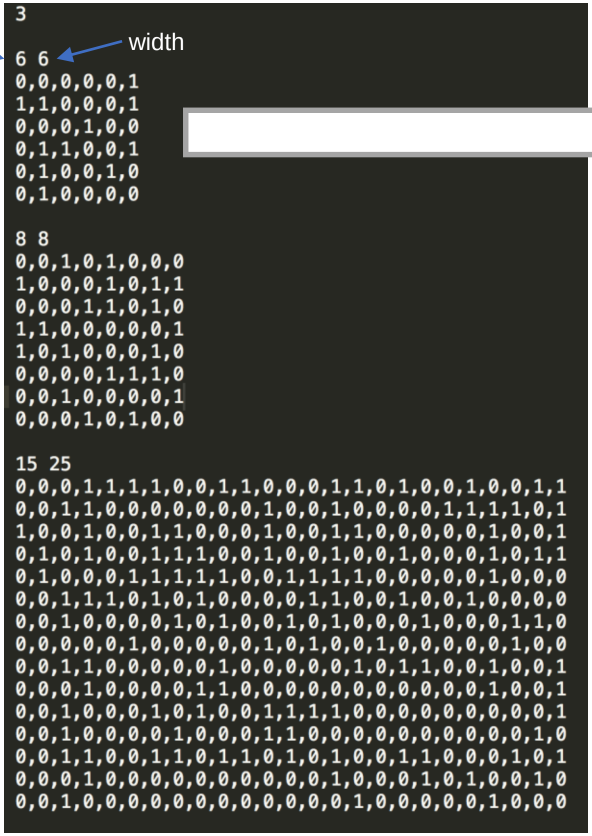


🟦	🟨	⬜	⬛	⬛	⬛	⬛	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬛	⬛	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬛	⬛
🟨	⬜	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬛	⬛	⬛	⬛	⬜	⬛
⬛	⬜	⬜	⬛	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬛	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬛
⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬛	⬛	⬜	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬜	⬛	⬜	⬛	⬛
⬜	⬛	⬜	⬜	⬜	⬛	⬛	⬛	⬛	⬛	⬜	⬜	⬛	⬛	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜
⬜	⬜	⬛	⬛	⬛	⬜	⬛	⬜	⬛	⬜	⬜	⬜	⬜	⬛	⬛	⬜	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬜	⬜
⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬜	⬛	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬛	⬛	⬜
⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜
⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬛	⬛	⬜	⬜	⬛	⬜	⬜	⬛
⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬛
⬜	⬜	⬛	⬜	⬜	⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬛	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛
⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜
⬜	⬜	⬛	⬛	⬜	⬜	⬛	⬛	⬜	⬛	⬛	⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬛	⬜	⬜	⬜	⬛	⬜	⬛
⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬛	⬜	⬛	⬜	⬜	⬛	⬜
⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜	⬜	⬜	⬛	⬜	⬜	⬜

3. Detail

- ## Input

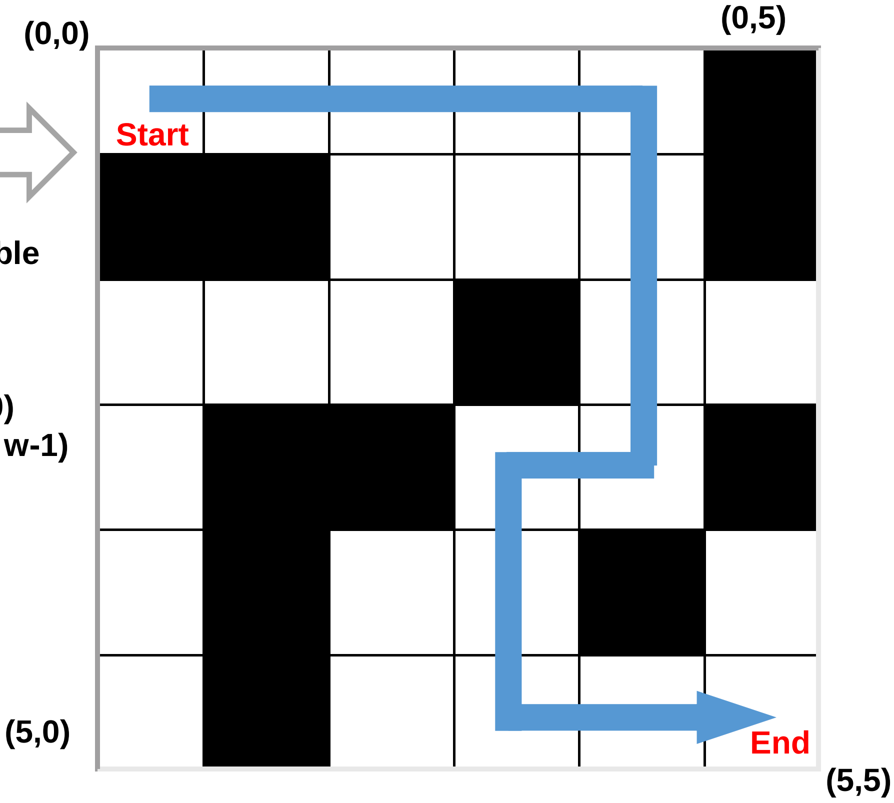
The first line has one integer, where number of mazes.
Each maze text has height, width and maze values. (height ≤ 30, width ≤ 30) Height and width separated by a single space, and each maze values separated by comma.



```
3

6 6
0,0,0,0,0,1
1,1,0,0,0,1
0,0,0,1,0,0
0,1,1,0,0,1
0,1,0,0,1,0
0,1,0,0,0,0

8 8
0,0,1,0,1,0,0,0
1,0,0,0,1,0,1,1
0,0,0,1,1,0,1,0
1,1,0,0,0,0,0,1
1,0,1,0,0,0,1,0
0,0,0,0,1,1,1,0
0,0,1,0,0,0,0,1
0,0,0,1,0,1,0,0

15 25
0,0,0,1,1,1,1,0,0,1,1,0,0,0,1,1,0,1,0,0,1,0,0,1,1
0,0,1,1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,1,1,1,1,0,1
1,0,0,1,0,0,1,1,0,0,0,1,0,0,1,1,0,0,0,0,0,1,0,0,1
0,1,0,1,0,0,1,1,1,0,0,1,0,0,1,0,0,1,0,0,0,1,0,1,1
0,1,0,0,0,1,1,1,1,1,0,0,1,1,1,1,0,0,0,0,0,1,0,0,0
0,0,1,1,1,0,1,0,1,0,0,0,0,1,1,0,0,1,0,0,1,0,0,0,0
0,0,1,0,0,0,0,1,0,1,0,0,1,0,1,0,0,0,1,0,0,0,1,1,0
0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,0,1,0,0,0,0,0,1,0,0
0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,0,1,1,0,0,1,0,0,1
0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1
0,0,1,0,0,0,1,0,1,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,1
0,0,1,0,0,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,0
0,0,1,1,0,0,1,1,0,1,1,0,1,0,1,0,0,1,1,0,0,0,1,0,1
0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0
0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0
```

0(white): passable node
1(black): wall

Start node: (0, 0)
End node: (h-1, w-1)

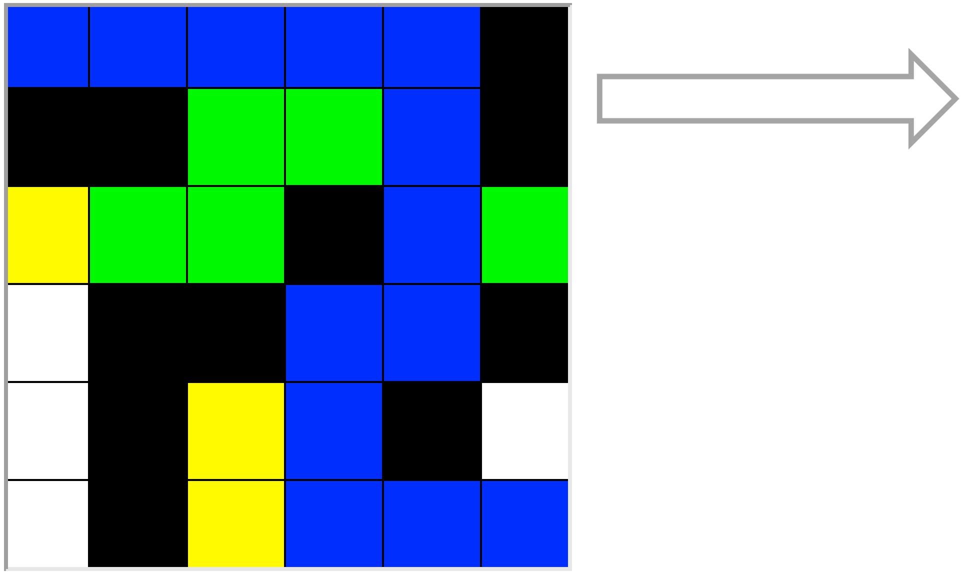


<Sample Input>

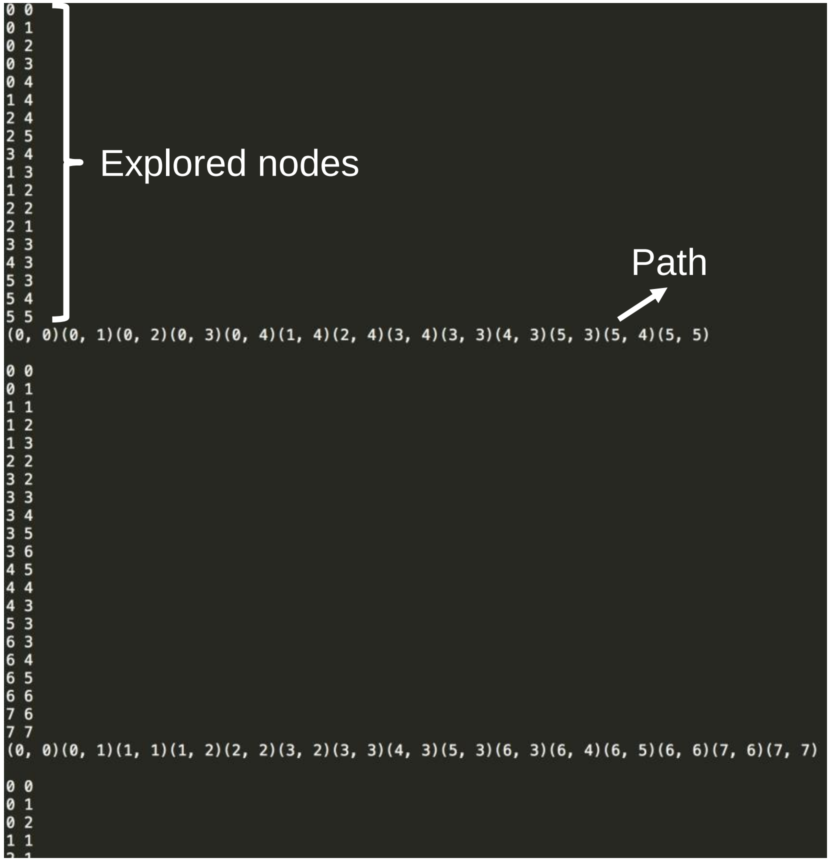
3. Detail

- **Output**

Print out all the explored nodes coordinates(each coordinate separated by a single space) in the order per line, followed by path nodes of optimal solution, as shown in the Sample Output(there is no space between each node coordinates in the path).



Blue: path, explored node
Yellow: candidate node to explore(but not explored)
Green: explored node
(Explored nodes are blue and green nodes)



```
0 0
0 1
0 2
0 3
0 4
1 4
2 4
2 5
3 4
1 3
1 2
2 2
2 1
3 3
4 3
5 3
5 4
5 5
(0, 0)(0, 1)(0, 2)(0, 3)(0, 4)(1, 4)(2, 4)(3, 4)(3, 3)(4, 3)(5, 3)(5, 4)(5, 5)

0 0
0 1
1 1
1 2
1 3
2 2
3 2
3 3
3 4
3 5
3 6
4 5
4 4
4 3
5 3
6 3
6 4
6 5
6 6
7 6
7 7
(0, 0)(0, 1)(1, 1)(1, 2)(2, 2)(3, 2)(3, 3)(4, 3)(5, 3)(6, 3)(6, 4)(6, 5)(6, 6)(7, 6)(7, 7)

0 0
0 1
0 2
1 1
2 1
```

Explored nodes

Path

<Sample Output>

3. Detail

- **Additional Assignment(Optional)**
  - Find the solution for given mazes using other search algorithm.
  - DFS, BFS, UCS… etc
  - If you using DFS algorithm, you do not need to find the optimal solution.
  - Each search algorithm +5pts
  - If you come up with your own search algorithm +10

3. Detail

- **Summary**

Your program is to read the input text file(input.txt) and find optimal solution(path). Then, print the result in output text file(output.txt) according to the output format.

If you have done additional assignment, submit one python code for each search algorithm. Each code print the result in output text file(output.txt) according to the output format.

We will test your code with various mazes.
The explored nodes and path in output text file do not have to be the same as our results, but the path must be the optimal solution.

4. Submission

- **Deliverables: 2013147xxx_1.zip**

- **Must include**
  - **Astar.py (Your code with detail comments)**
  - **DFS.py, BFS.py.. etc (additional assignment)**
  - **Other codes (If you necessary)**

5. Grading environment & Directions

- Language: Python
- We grade your score in Linux(Ubuntu 16.04)
- Python3 (>= 3.5.2)

- This is an individual project
- You should follow the input/output format

- Never copy code
- You will get 0 points if you cheat

6. Grading policy

- One maze has a total of 10pts.
- Find a optimal solution: 5pts
- Depending on the number of explored nodes: 0~5pts

7. Due Date

- Due Date: 27/March/2018 23:59:00 KST
- Delay Policy: -15pts per day

Pleae use YSCEC Q&A board to leave your question.