

# Introduction to Artificial Intelligence

## Assignment 1: The Matrix Universe

### Deadline

Week 10 (01/11/24 11:59 PM)

### Submission

1. Two source codes (same language) files have to be uploaded to Codeforces (one for each algorithm)
  - Codeforces handle doesn't have to be uploaded to Moodle if was sent during ITP/SSAD/TCS courses and not changed after (otherwise, contact course instructor)
2. Report describing each algorithm flow in plain English (not more than 3 paragraphs), statistical comparison among algorithms and PEAS description with respect to the Actor agent should be uploaded to Moodle. Name of the file should be like this *NameSurname.pdf* (For example, *IvanIvanov.pdf*).
  - Report should include graphical representation of maps that were impossible to solve. You can generate the maps directly through the code or hand draw them for the report after testing them on both algorithms
  - Report may include any interesting outcome/map worth highlighting

### Programming Language

Java (JDK 8 or 21) or C++ (C++17 or 20 or 23) or Python 3

### Requests

- The program must work, the code should be readable, well-structured and should contain comments in a language-specified format
- It has to be only one report *\*.pdf* file
- It is allowed to use only standard libraries
- NO extension of a deadline. Works sent after the deadline will NOT be evaluated
- Assignment is individual
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies

### Grading Criteria

- 60% for passing tests
- 10% for code readability and comments quality
- 30% for the well-structured and informative report (given working code)

## Task

*Neo*, a seemingly ordinary computer programmer, finds himself thrust into a world of virtual reality known as the *Matrix*. Trapped within this simulated reality, humanity is unknowingly controlled by machines. As Neo delves deeper into this digital prison, he discovers a shocking truth: the world he once believed to be real is merely a carefully constructed illusion.

Central to Neo's journey is the *Keymaker*, a mysterious figure who holds the key to unlocking the Matrix and freeing humanity. Neo must navigate the treacherous digital landscape, evading the relentless pursuit of *Agent Smith* and the *Sentinel*. Agent Smith, a powerful program created by the machines, is relentless in his pursuit of Neo, using his ability to replicate himself to overwhelm opponents. The Sentinel, a physical construct of the machine world, patrols the Matrix to ensure human compliance.

As Neo draws closer to his goal of finding the Keymaker, he discovers the existence of the *Backdoor Key*. This elusive tool, hidden within the Matrix, could be a game-changer. If Neo can acquire the Backdoor Key, he may be able to permanently disable Agent Smith but it will take him out to the real world, Sentinel will become even stronger and will chase Neo more. However, seeking the Backdoor Key could divert Neo from his primary mission and increase his chances of being captured.

Neo is not alone in his struggle. *Trinity*, a skilled hacker and Neo's ally, is trying to assist him from the real world. Although she can never go to the matrix game, she can provide him with crucial information about the enemy's movements and perception zones, helping him avoid detection and anticipate their actions.

With or without Trinity's help, Neo with his own determination must navigate the treacherous world of the Matrix and overcome the challenges that lie ahead. The fate of humanity depends on his success. As Neo fights for freedom, he must confront the terrifying reality of the machine world and the ultimate question: can a single individual truly make a difference?

Your task is to help Neo to find the shortest path to Keymaker by using A\* and Backtracking without being killed!







	0	1	2	3	4	5	6	7	8
0									
1									
2									
3									
4									
5									
6									
7									
8									

Figure 1: Map Example

## Map Details

The environment consists of a 9x9 grid representing different sectors of the Matrix. Neo starts in the top left corner which is always a safe cell (0, 0), while the Keymaker is located at a known position also in a safe unoccupied cell. Enemies represented by entities of Agent Smith and Sentinel are randomly distributed in unoccupied cells and only one per cell. Backdoor Key is also located in a safe unoccupied cell. One of the possible map representations is provided in Figure 1.

## Neo

- His goal is to reach the Keymaker while avoiding entities of Agent Smith and Sentinel and their killing zones. The killing zones of enemies suit their perception zones. Neo perceives the environment according to the Moore neighborhood of range 1 or 2 depending on the perception variant. However, in both variants Neo can only move one step at a time either vertically or horizontally. Neo is also called Actor in this assignment context. In Figure 2 the first variant of perception containing 8 consecutive cells is provided.

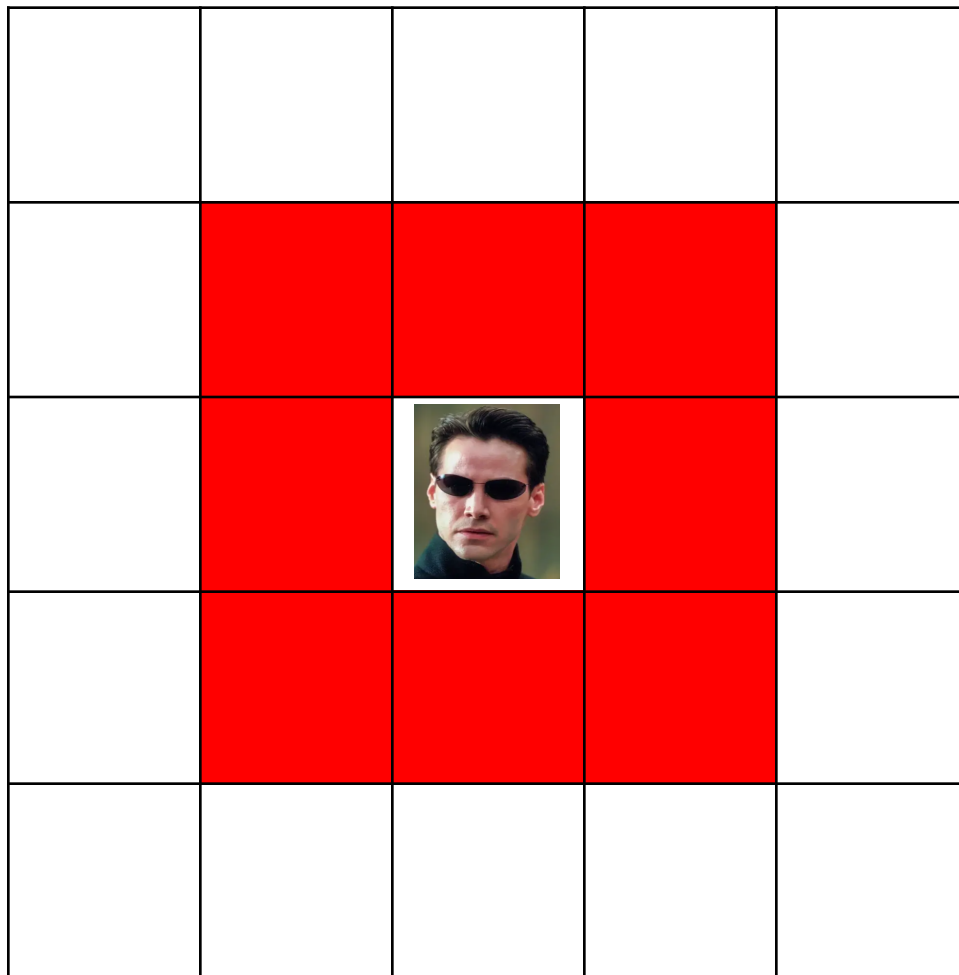


Figure 2: Neo's Perception Zones (Variant 1)

## Enemies

- **Agent Smith**

There are from 0 to 3 entities of Agent Smith in the map. Each Agent Smith can perceive Neo according to the Moore neighborhood of range 1 which contains 8 consecutive cells (Figure 3). Location of Neo in the perception zone of Agent Smith leads to lethal outcomes for Neo, because Agent Smith has a gun. The positions of Agent Smith entities are unknown to Neo.

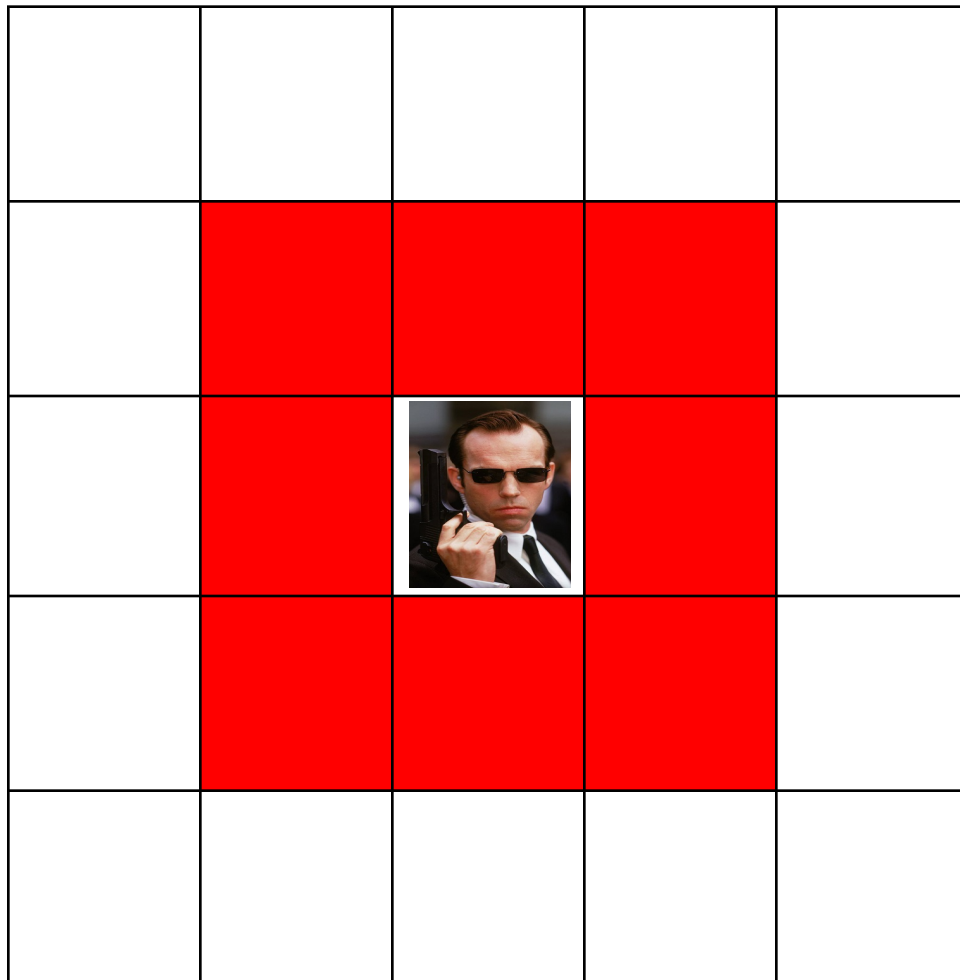


Figure 3: Agent Smith's Perception Zones

- **Sentinel**

The number of Sentinels is 0 or 1. Sentinel has a perception range described by von Neumann neighborhood of range 1 (Figure 4). It punches Neo by its tentacles if the latter enters the former's predefined perception range. The position of Sentinel is unknown to Neo.

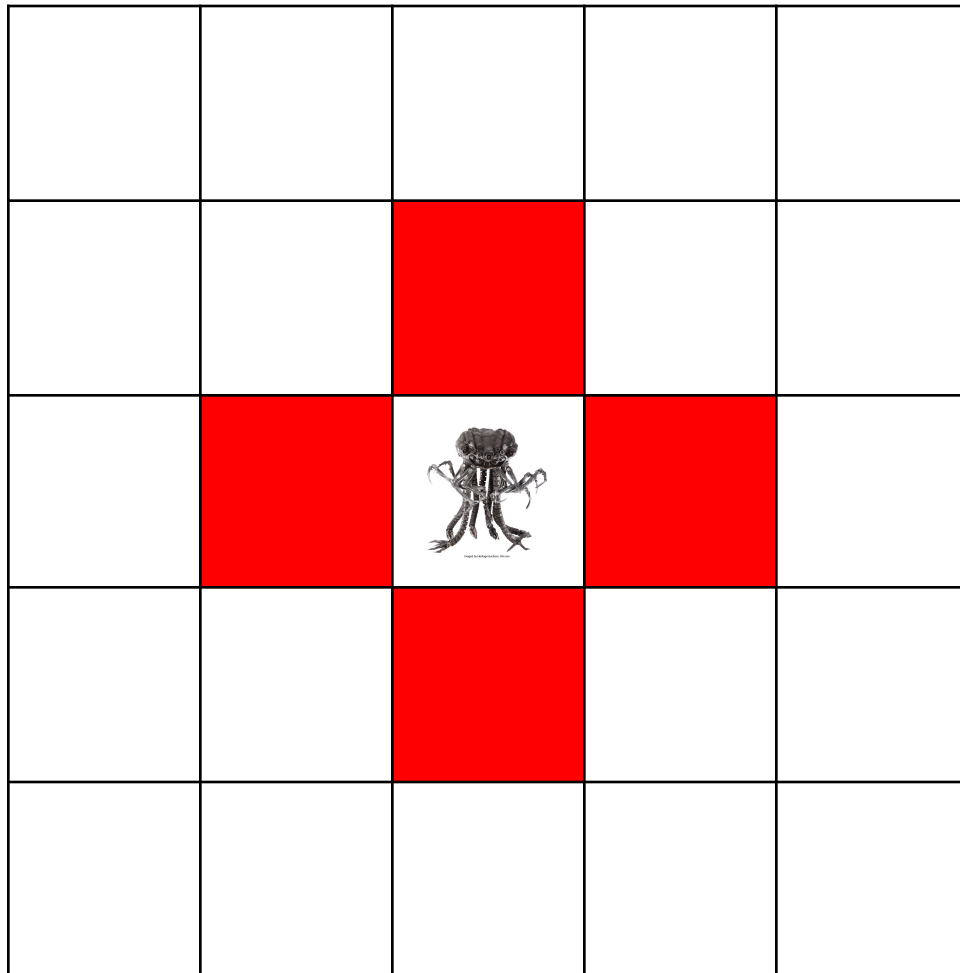


Figure 4: Sentinel's Perception Zones

## Keymaker

Neo knows the position of the Keymaker (Figure 5), who is vital for advancing through the Matrix.



Figure 5: Keymaker

## Backdoor Key

If Neo finds the Backdoor Key (Figure 6), it grants him special abilities:

- **Disable all entities of Agent Smith:** Permanently disables Agent Smith, because Neo will leave the Matrix.
- **Visibility Increment of Sentinel:** Increase perception zone (extending of tentacles) of Sentinel till von Neumann neighborhood of range 2 (Figure 7), because this machine wants Neo to be inside the Matrix.

Neo doesn't know the location of the Backdoor Key and he may find it to utilize these advantages. However, this is not an obligation.



Figure 6: Backdoor key

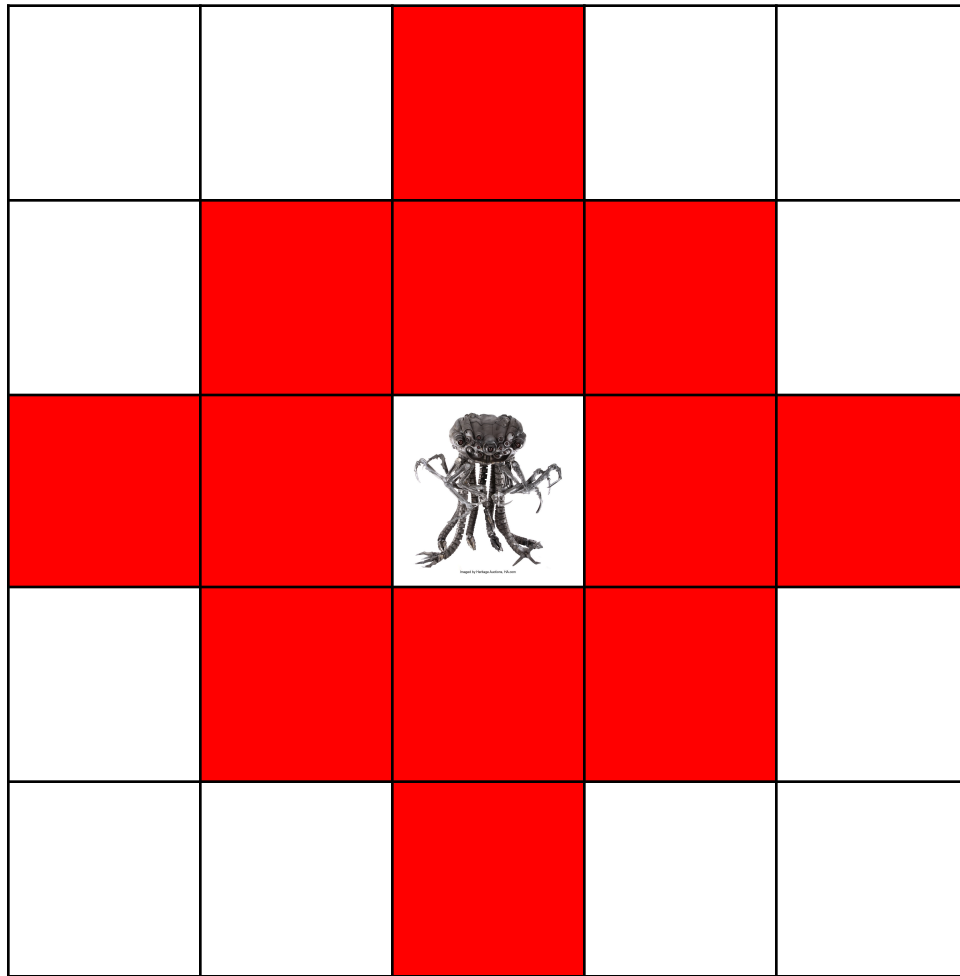


Figure 7: Sentinel's new Perception Zones

## Algorithms

- A backtracking search
- A\*

You are allowed to use modified versions of algorithms, do not forget to describe modifications in the report, if there will be any. Note that submission of the same or very close code for both algorithms will be considered as a cheating case.

### Variant 1

You can perceive the enemy and its perception zones in Moore neighborhood cells of range 1. This can be done if you are standing next to their perception zone or even next to them in case of some enemies, shown below in Figures below. In Figure 8-a, you are able to perceive 3 orange cells, in Figure 8-b only 1. Orange cells indicate the Actor's perception of the enemy and its zones



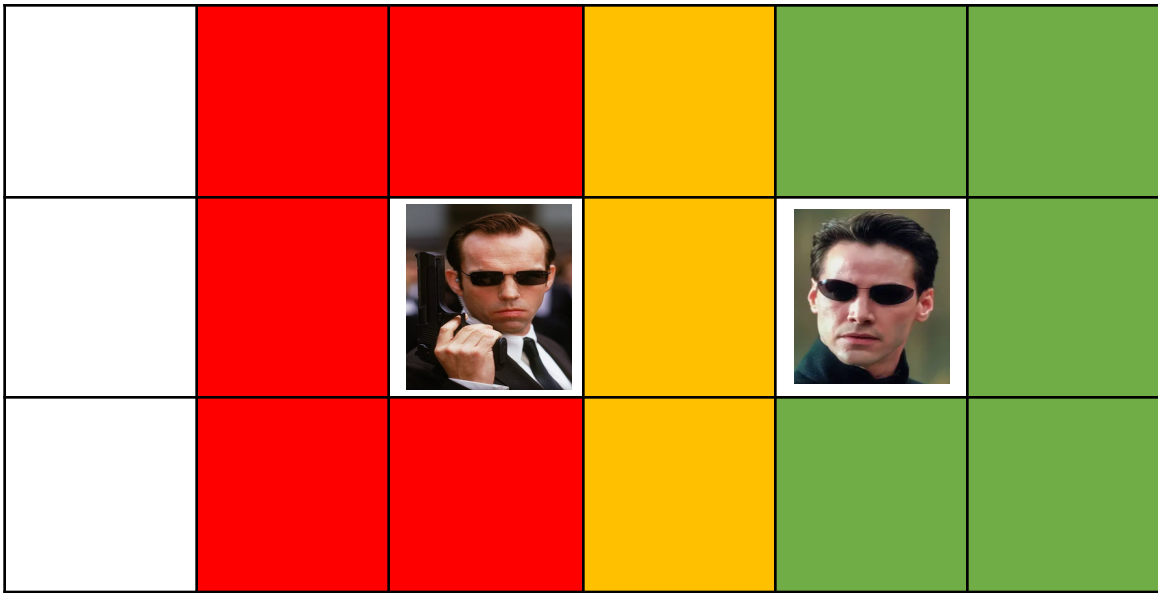


Figure 8-a: Neo’s Perception Zones

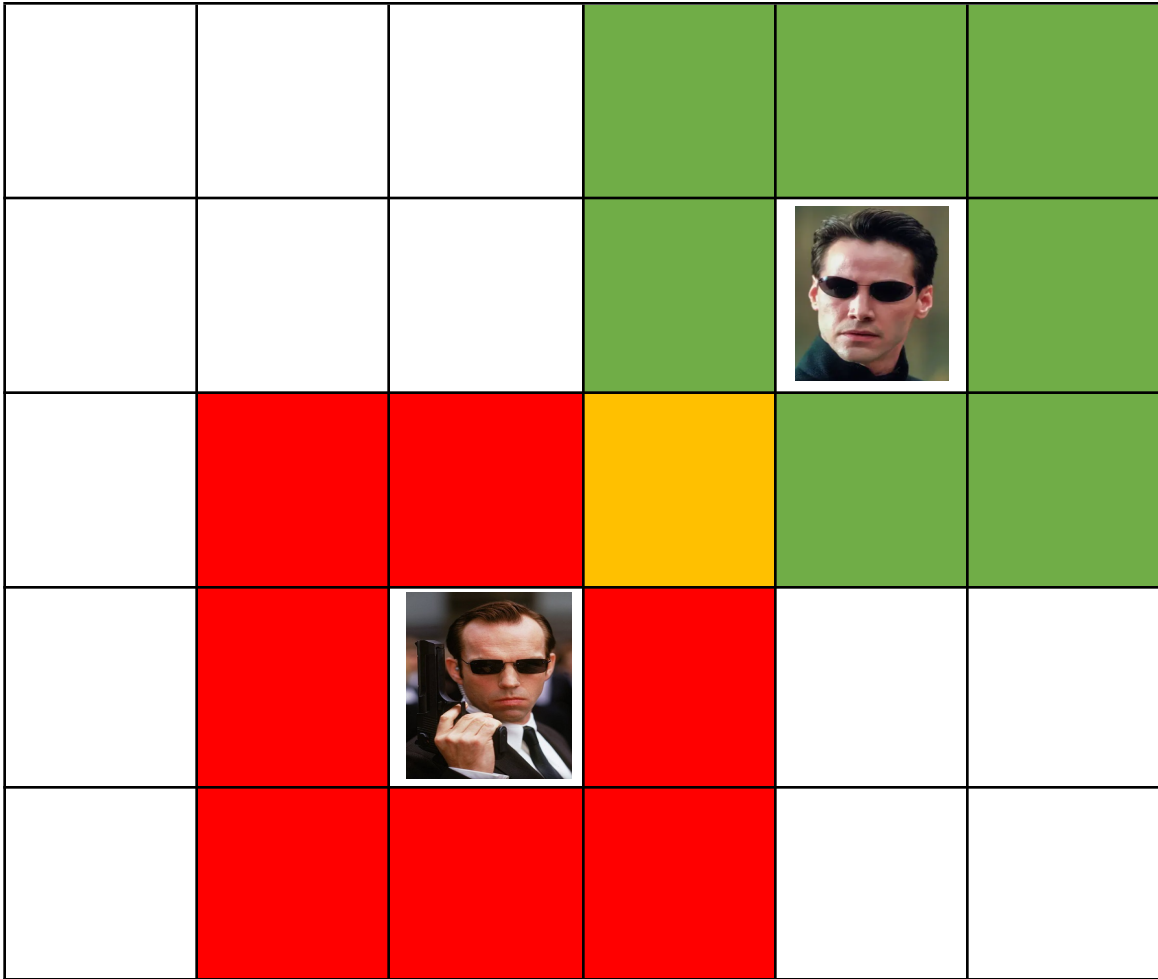


Figure 8-b: Neo’s Perception Zones

### Variant 2

Trinity (Figure 9) is assisting you from the real world and trying to help you throw. Because of her you can perceive the enemy and its perception zones better in Moore neighborhood cells of range 2 (Figure 10). In Figure 11-a, you are able to perceive 6 orange cells, in Figure 11-b only 4. Orange cells indicate the Actor's perception of the enemy and its zones



Figure 9: Trinity

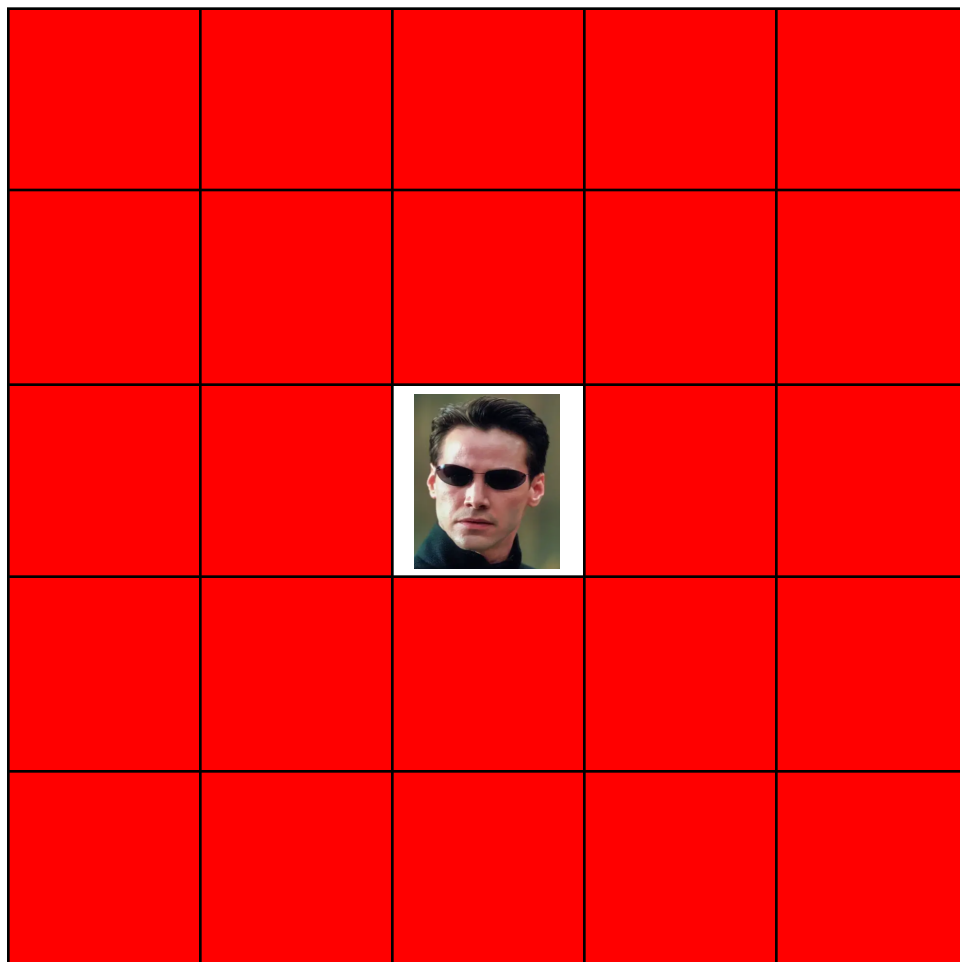


Figure 10: Neo's Perception Zones

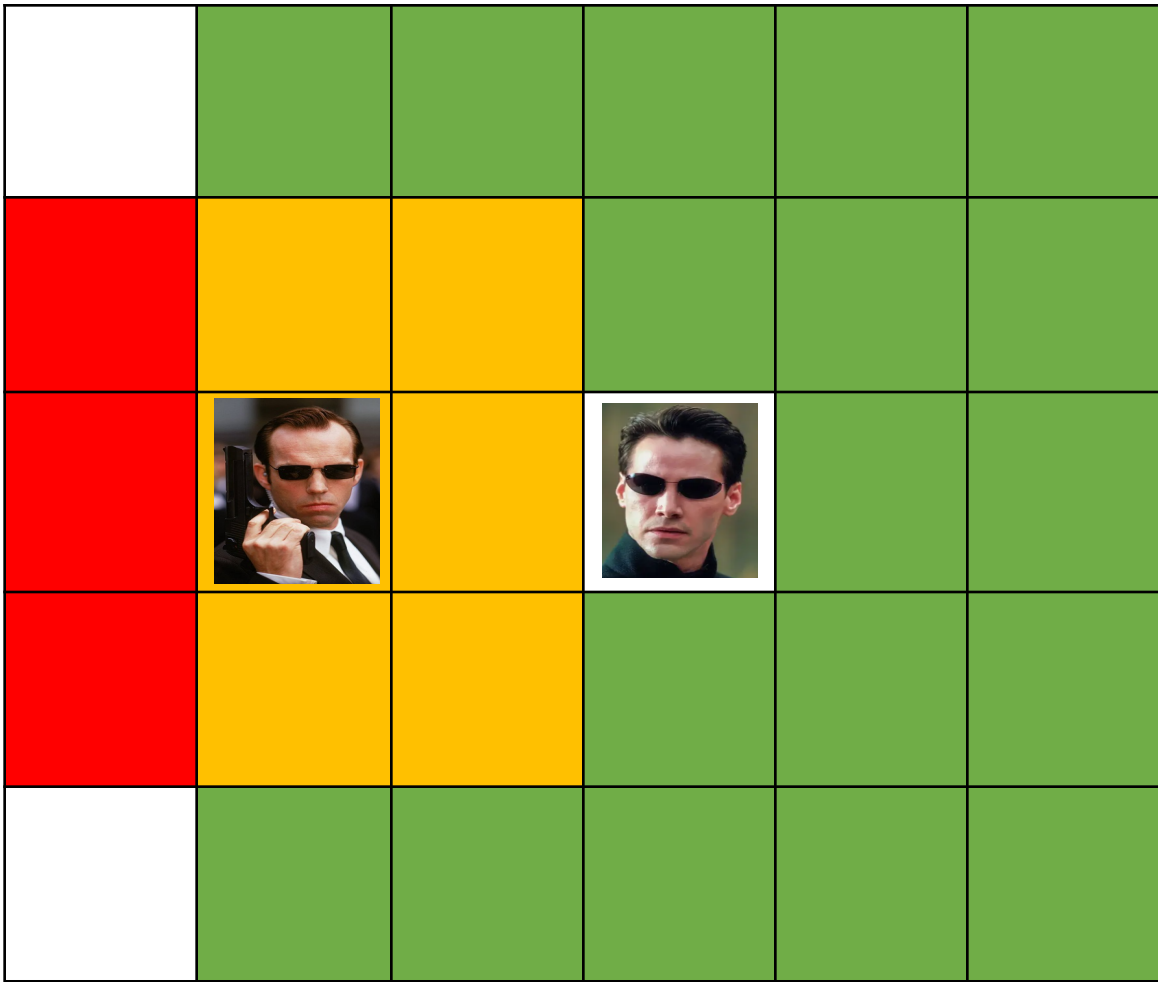


Figure 11-a: Neo's Perception Zones

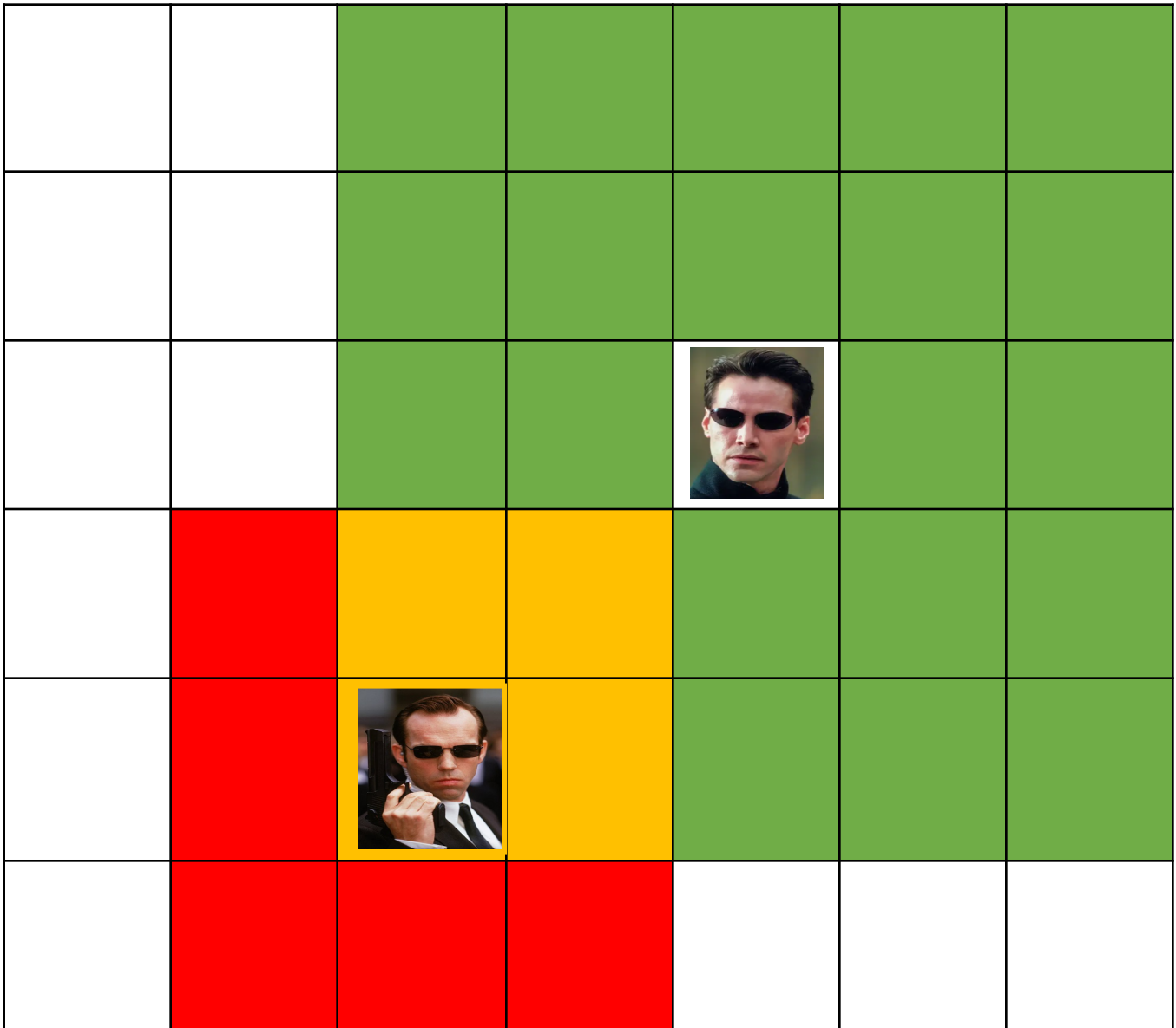


Figure 11-b: Neo's Perception Zones

## Inputs and Outputs

You will be communicating with Codeforces interactor through standard inputs and outputs in your code, and the interactor checks it.

1. Your code should start by accepting an integer (1 or 2) that represents the variant number for Neo's perception scenario. Then you should accept a pair of integers representing the coordinates  $(x, y)$  of the Keymaker.
2. After that, you should start communicating with the interactor to explore the map. You always start at position  $(0, 0)$ . The interactor accepts one of 2 arguments

a. **m**  $x$   $y$ :

- **m** is given to tell the interactor that you want to make a **move**
- **x** and **y** are the coordinates you want to move to, where **x** stands for rows, and **y** stand for columns (value in range  $[0; 8]$ )
- The interactor will then return to you the dangers around you if there are any. If the cell is dangerous (contains any enemy or its perception zone), you will receive the coordinates of this cell and the danger type from the interactor.
- Note that you can only move to the allowed neighboring cells; you cannot teleport through the map. For example, if you are currently at position  $(3,4)$ , you cannot send the command **m 5 6** (will lead to a failed test)
- Example:

Input example: **m 3 4**

Output example (given you just moved to  $(3,4)$ ):

3

2 3 P

2 4 S

3 3 P

where:

- First row is the number of surrounding dangers (0 if no dangers)
- Numbers in the other rows (if any) are the coordinates of the agents followed by the type of agent:
  - P** is Perception zone of enemy
  - A** for Agent Smith
  - B** for Backdoor Key
  - S** for Sentinel
  - K** for Keymaker
  - N** for Neo.

- b. When your algorithm comes to an end you send **e** followed by integer
  - **e** represents that you reached the **end**

- -1 for valid unsolvable map
- Any integer greater than 0 for the length of the shortest path to the Keymaker
- Note that invalid maps won't be provided in Codeforces tests. However, you are asked to think about it carefully for the report statistics

You always start in coordinates (0,0) and given the surroundings in your perception zone. When you die, the game is over for the current test. For the next test you go back to (0,0) and start all over again.

More details about interaction can be found on Codeforces.

## Statistical Analysis

Comparison of algorithms through statistical analysis should be based upon 1000 test maps generated by you. The statistics should provide the mean, mode, median and standard deviation for the execution time, number of wins and number of losses. Also, the percentage of wins and losses should be provided. Statistical analysis is required for each map, comparison between: **Backtracking** compared to **A\*** for both variants. Comparison of the algorithms requires data in tables and further discussion about it.