Ahmed Nouralla – B19-02 – AI Assignment 1 Report.

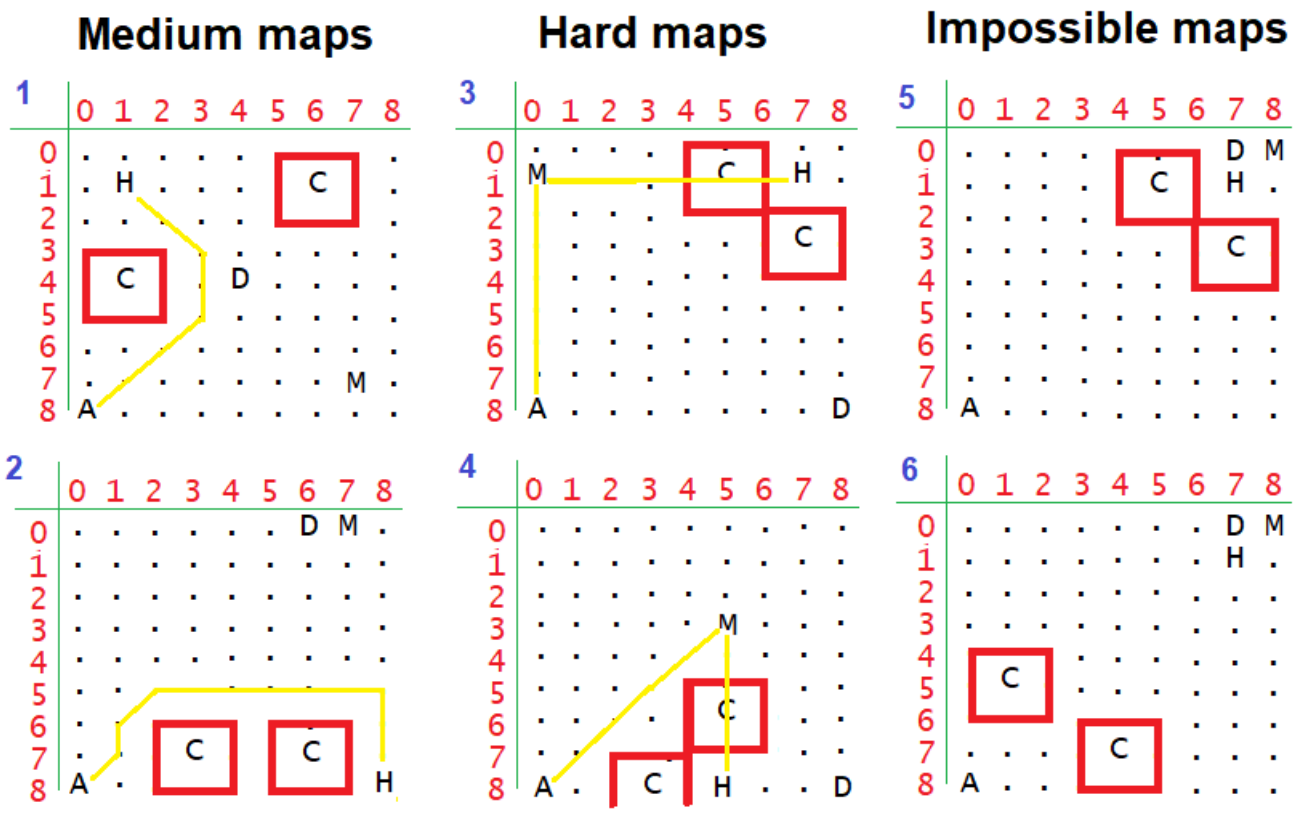
- Source code files are included in the same archive with this report.
- **Algorithms' description:**
  - **Backtracking search** (backtracking.pl)**:**
    - The idea was to use a recursive predicate go(StepCount, Path, NextMove, Protected) that explores the (auto-generated) map, by trying all possible valid paths while counting the number of steps along the way, minimizing StepCount, keeping the state of actor (whether he is Protected from covid or not), and storing the Path list.
    - Since the number of recursive calls will be huge (even for 9*9 lattice, the unguided search is expensive and better algorithms exist for shortest path problems), some optimizations were implemented to make it faster for typical cases, but the upper-bound complexity didn't change.
    - Example of such optimizations was to try the recursive calls that are more likely to get the actor home first, by realizing the position vector from the actor current location to home and guiding the search.
  - **A* search** (astar. pl)
    - The algorithm uses the diagonal distance heuristic $\max(|actor_x - home_x|, |actor_y - home_y|)$ to determine the best next move (because the actor can move in max. 8 directions)
    - Since the problem is not pure shortest path, and is more like a maze-door-keys problem, the algorithm is modified to consider 3 possible cases.
      - Actor goes around "covid" to reach home.
      - Actor goes to the doctor (while avoiding covid), then directly to home
      - Actor goes to grab the mask (while avoiding covid), then directly to home.
    - After determining the 3 potential ways for the actor to reach home, the algorithm choses a one that has the shortest overall number of steps

- **PEAS description with respect to the actor agent.**
  - **Performance measure:** the number of steps needed to reach home, whether the actors can reach it or not.
  - **Environment:** 9*9 square lattice, representing physical spots.
  - **Actuators:** the actor can **move** (legs) horizontally, vertically, and diagonally.
  - **Sensors:** the actor can **perceive** (eyes) objects around him, from different distances.

- **Statistical analysis** (for random samples of maps)
  - Note that variants didn't introduce any output difference; the actor perceiving covid from a larger distance may affect his decision of going in a certain direction, but will never affect the final shortest path length, as it's unique.
  - Running 10 randomly generated maps and collecting the running time results (in seconds)
    - Check this link for a detailed output.
    - **Backtracking search:**
      - **Mean time** ≅ 6.384
      - **Median time** ≅ 2.966
      - **Standard deviation** ≅ 8.697
      - **Generally:** the process is almost always CPU intensive (93% - 100% CPU utilization), due to the large number of recursive calls.
    - **A* search:**
      - **Mean running time:** 1.444 seconds
      - **Median running time:** ≅ 0.4665
      - **Standard deviation:** 2.975
      - **Generally:** the process is usually not-CPU intensive, since the algorithm is efficient and the map is of dimensions 9*9

#	A*	BT
1	0.418	1.956
2	0.419	3.187
3	0.480	4.288
4	0.461	0.615
5	0.472	0.623
6	0.443	7.427
7	10.369	11.072
8	0.410	30.725
9	0.485	1.206
10	0.483	2.745
Σ	**14.44**	**63.84**

- **Graphical representation for custom maps used for testing:**



- **Statistical analysis for custom maps:**
  - **Map 1:**
    - Backtracking: success, 11,837,908 inferences, 0.531 CPU in 0.545 seconds
    - A*: success, 31,727 inferences, 0.000 CPU in 0.005 seconds
  - **Map 2:**
    - Backtracking: success, 1,816,199,138 inferences, 128.625 CPU in 128.802 seconds
    - A*: success, 430,249 inferences, 0.125 CPU in 0.129 seconds
  - **Map 3:**
    - Backtracking: will eventually succeed, although taking a lot of time.
    - A*: runs into an infinite loop
  - **Map 4:**
    - Backtracking: will eventually succeed, although taking a lot of time.
    - A*: success, 73,289,799 inferences, 36.188 CPU in 36.305 seconds
  - **Map 5:**
    - Backtracking: will eventually terminate with answer (no path exists), although taking a lot of time, since it has to do the worst-case number of recursive calls
    - A*: runs into an infinite loop (no solution exists).
  - **Map 6:**
    - Backtracking: terminates quickly (38,192 inferences, 0.016 CPU in 0.017 seconds) with answer (no path exists), since all recursive calls will be blocked early.
    - A*: terminates with message (no solution exists).

A* with Map 3
- Although there is a solution, A* will try to consider the option of a direct way home, which doesn't exist.
- This can be easily solved by stating a time-limit for the algorithm to run before considering other options, or by explicitly checking the edge case when a non-covid element is surrounded by covid.