# MAZE ON FIRE

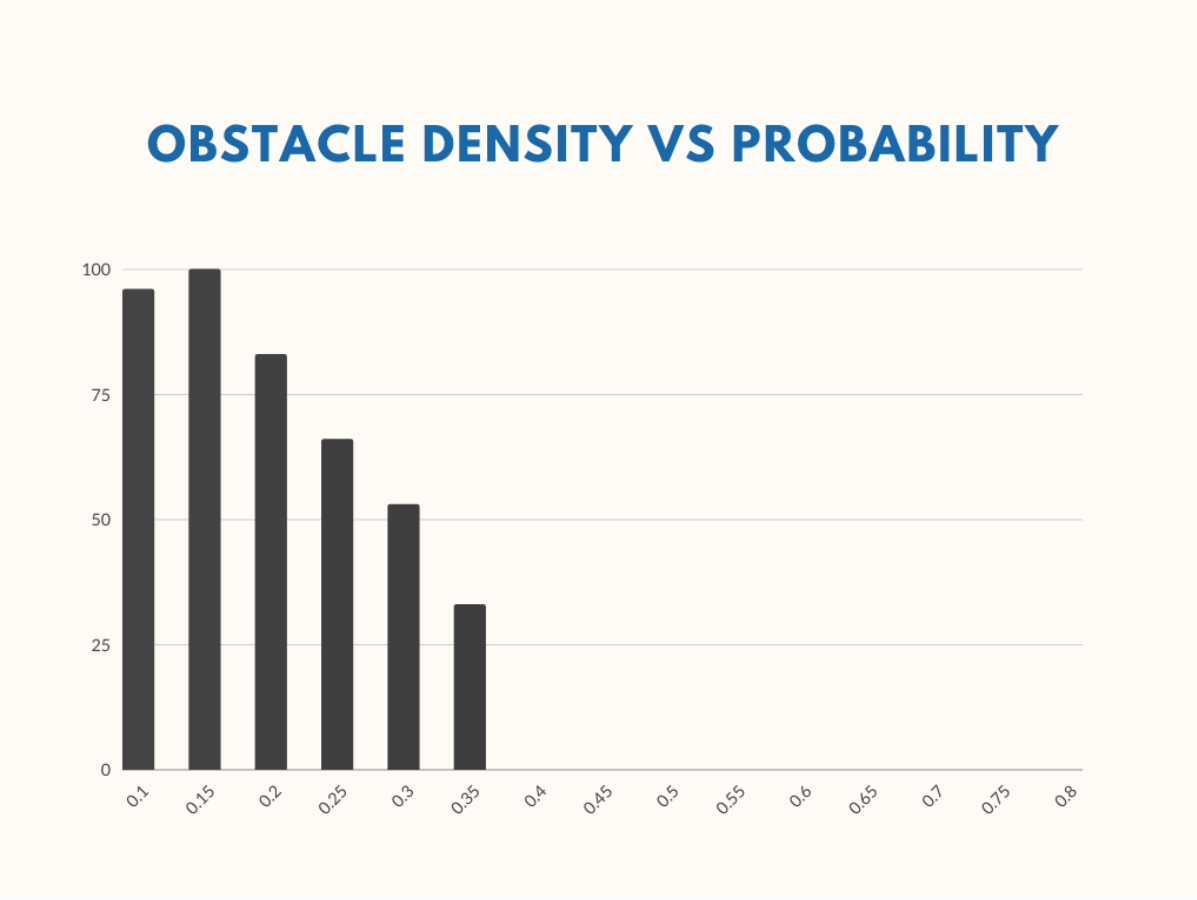
Cs 440 Project 1
Sangkyun Kim sk1998 - 05 , Tenzin Norden tn266 - 08

https://github.com/Norden-Tenzin/440ArtificialIntelligence.git

We split the work 50 50 and worked on the entire project together. We had daily meetings to discuss our progress. We used github as a collaboration tool. We would talk about the issue at hand before tackling it and try to break down the problem into smaller more easy to solve problems.

## Problem2:

BFS needs to visit almost every block in the maze( around 1700 visited block in 50*50 maze) to find the exit route. However, DFS visits needs smaller number of visiting (around 500 visited block in 50*50)to find the path. BFS requires more memory and takes more time.



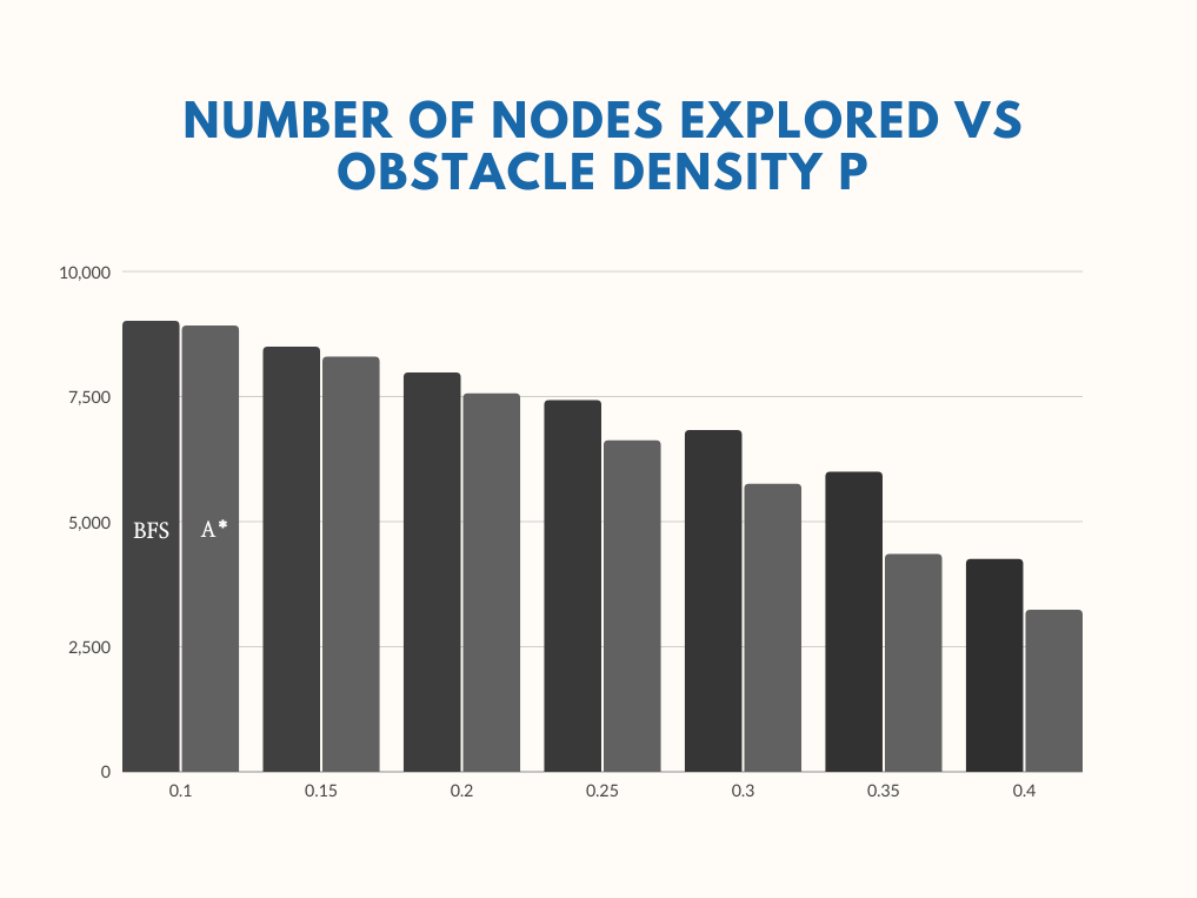
X - axis: Probability of exit the maze

Y - axis: Obstacle density P

Maze size = 100 * 100

# Problem 3:

There will be no difference in the number of nodes explored because if no path is found to complete the maze they will explore all nodes available to them.




X - axis : Number of visited node

Y - axis: Obstacle density P

Maze size = 100 * 100

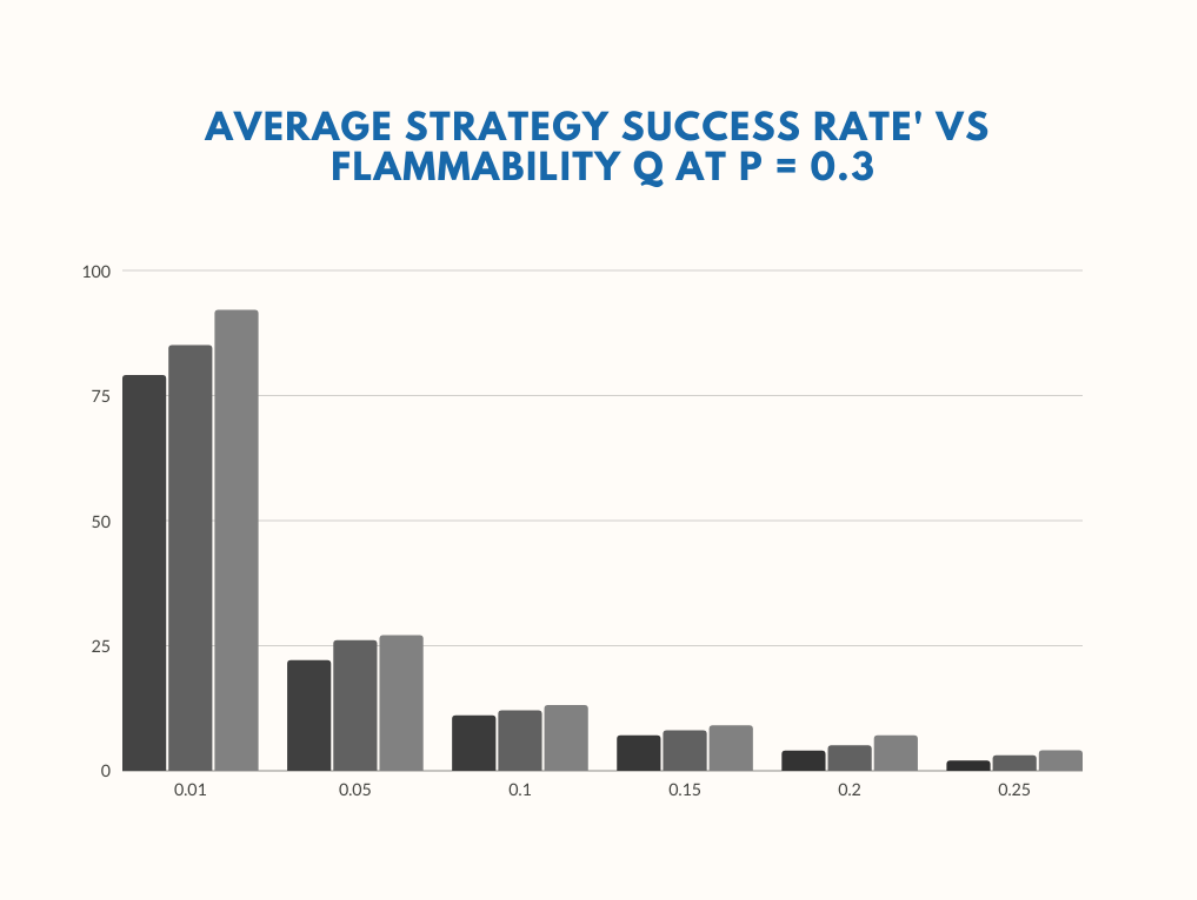
# Problem 4

- DFS: 190 * 190 (p = 0.3)
- BFS: 180 * 180 ( p = 0.3)
- A*: 200 * 200 ( p = 0.3)

# Problem 5

For Strategy 3 we built upon strategy 2 by adding another heuristic which calculated the distance between the agent and the initial fire position. Since the algorithm knows the location of the initial fire block it  avoids that section assuming that it will catch on fire.

# Problem 6




X -axis: Success rate (%)
Y- axis: Flammability Q

Maze size = 50 * 50

Black: Strat1
Dark Grey:  strat2
Grey : strat3

They perform differently when fire is spreading slowly. We believe this happens because strategy 1 and 2 aren't aware of the fire's location leading them to run straight into it, on the other hand our strategy 3 being aware performs better than them in these situations.
They perform fairly similarly when fire is spreading rapidly. This happens as no matter what the algorithm can do the fire is just too fast and the agent can't escape.

# Problem 7

If we had unlimited computational resources at our disposal we would look to find the possible future fire paths and take that into account along with our strategy 3. Along with this we would simulate future possibilities at each step and compare them and pick the route with the best survivability.

# Problem 8

In every step we would have liked to find the closest fire block to the agent's position and modify the heuristic value based on that distance to give the algorithm better awareness and of fire while still being fast.