Name	ID
Osama Abutaha	S23108673

Discussion Questions:

1. Why does the current implementation use StackFrontier instead of QueueFrontier?

In this line (127) we can choose either StackFrontier() or QueueFrontier() – Now it's StackFrontier() which is DFS, and we use it because it uses less momery.

2. What would happen if you changed line 127 to use QueueFrontier()?
   It will use BFS(Queue) instead of DFS(Stack)

   • How would the exploration pattern change? In Queue it's will search widely not deeply

   • Would it find the same path? No, it may find a different path. BFS is guaranteed to find the shortest path.

   • Would it explore more or fewer states? It depends on the maze, but till now it explored fewer states

3. Is DFS guaranteed to find the shortest path? Why or why not? DFS(Stack) not guaranteed to find the shortest path and may explore very long path, because it goes deeply.

4. How does the number of explored states differ between maze1.txt and maze2.txt?

   • Run both and compare the "States Explored" output

The maze	DFS	BFS	Number of States Explored
1	11	11	
2	194	77	

   • Why is there a difference? Because each one has different maze structures and have different goal.

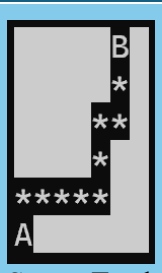
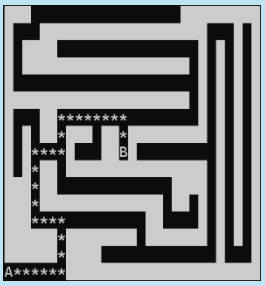
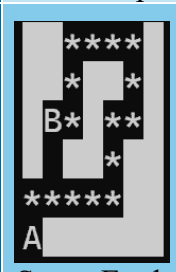
5. What would happen if we removed the explored set check?

   • Hint: Think about cycles in the maze

In some cases it might go infinitely (infinite loop), because it may visit the same cell many times without noticing that it was visited before.

Exercise 1: Run and Observe

1. Run the program with all three maze files
2. Record the number of states explored for each
3. Compare the solution paths in the generated images

Maze \ Algorithm	DFS
1	 States Explored: 11
2	 States Explored: 194
3	 States Explored: 17

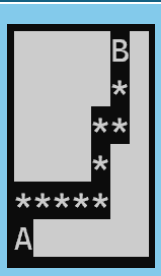
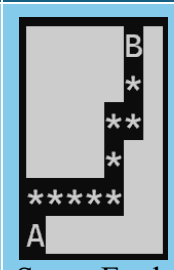
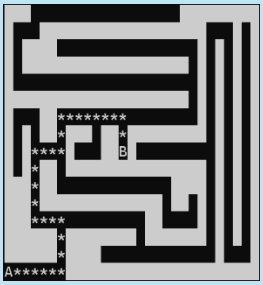
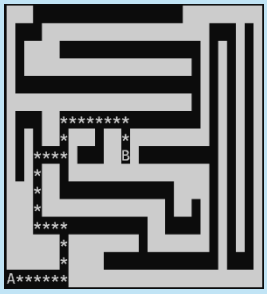
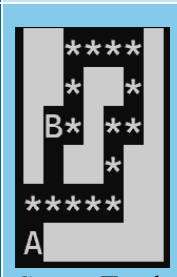
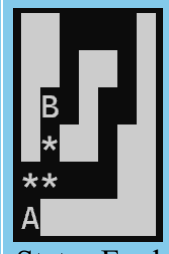
Exercise 2: Switch to BFS

1. Modify line 127 to use QueueFrontier() instead of StackFrontier()

```
1  frontier = QueueFrontier()
```

2. Run the program again with all maze files

3. Compare: states explored, path length, and path shape

Maze \ Algorithm	DFS	BFS
1	 States Explored: 11	 States Explored: 11
2	 States Explored: 194	 States Explored: 77
3	 States Explored: 17	 States Explored: 6

4. Document your findings

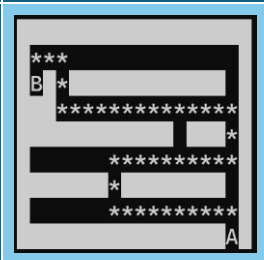
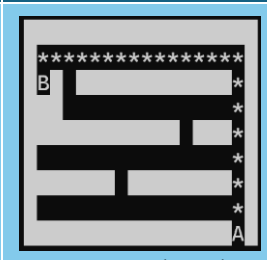
While both algorithms use different ways of searching, but I noticed that in maze 1 and 2 both DFS and BFS have the same result path regardless of the number of States Explored.

Exercise 3: Create Your Own Maze

1. Create a new file maze4.txt with your own maze design
2. Make sure it has exactly one A and one B
3. Test it with both DFS and BFS
4. Design a maze where DFS and BFS find different paths

Maze4.txt →

```
####################
#                  #
#B# ############ #
###              #
############ ### #
#                #
####### ######## #
#                #
#################A#
```

Maze \ Algorithm	DFS	BFS
4	 States Explored: 61	 States Explored: 70