# Maze Search: Arnav Krishnan

Maze:

**********

*--------*

*-******-*

*---*--*-*

***-*-**-*

*---*----*

*-******-*

*------*-*

******-*-*

*--------*

DFS Search:

**********

*S-------*

*o******-*

*ooo*--*-*

***o*-**-*

*ooo*----*

*o******-*

*oooooo*-*

******o*-*

*-----ooG*

BFS Search:

**********

*Soooooo*

*-******o*

*---*--*o*

***-*-**o*

*---*---o*

*-******o*

*------*o*

******-*o*

*-------G

The fundamental distinction between DFS and BFS lies in their approach to exploration, with the former employing a Stack while the latter utilizes a queue. While BFS offers advantages in terms of guaranteeing the shortest path, these benefits come at the cost of memory usage, particularly in scenarios where the graph necessitates many nodes being held in the queue simultaneously. This was seen in my results, with BFS indeed finding the shortest path, whereas DFS opted for a longer, meandering path that I intentionally made knowing that DFS would opt for it.

Although not a key part of the write-up, I would like to include this strip from a comic I frequently read:

This xkcd comic is relevant as it highlights the difference between breadth and depth first searches.