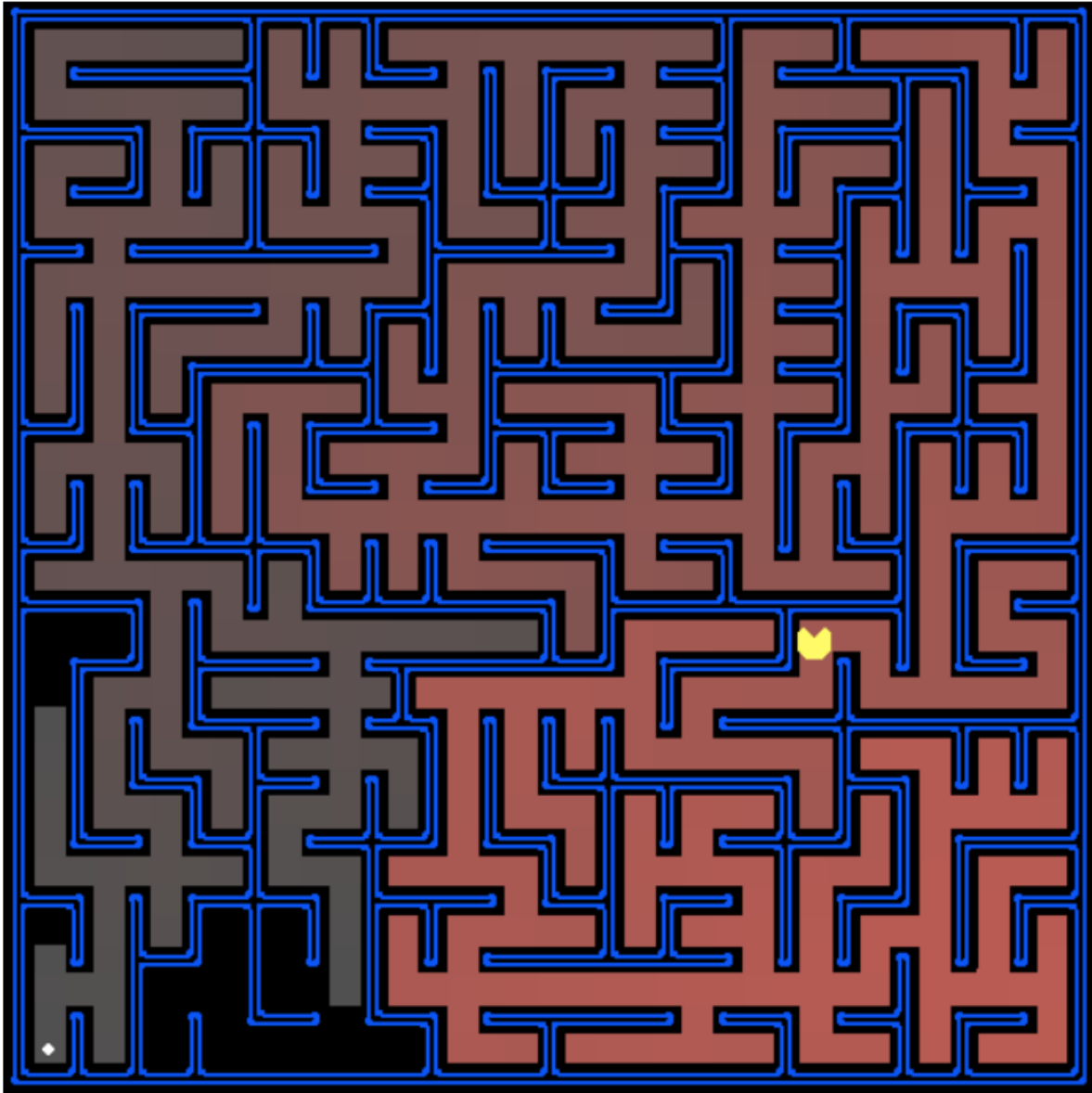


Project 1: Search



All those colored walls,
Mazes give Pacman the blues,
So teach him to search.

Introduction

In this project, your Pacman agent will find paths through his maze world, both to reach a particular location and to collect food efficiently. You will build general search algorithms and apply them to Pacman scenarios.

Question 1 (2 points): Finding a Fixed Food Dot using Depth First Search

In required project, you have to find a fully implemented search agent, which plans out a path through Pacman's world and then executes that path step-by-step. The search algorithms for formulating a plan are not implemented – that's your job.

Important note: All your search functions need to return a list of *actions* that will lead the agent from the start to the goal. These actions all must be legal moves (valid directions, no moving through walls).

Hint: Each algorithm is very similar. Algorithms for DFS, BFS, UCS, and A* differ only in the details of how the fringe is managed. So, concentrate on getting DFS right and the rest should be relatively straightforward. Indeed, one possible implementation requires only a single generic search method which is configured with an algorithm-specific queuing strategy. (Your implementation need *not* be of this form to receive full credit).

Question 2 (2 points): Finding a Fixed Food Dot Breadth First Search

Does BFS find a least cost solution? If not, check your implementation.

Question 3 (2 points): Finding a Fixed Food Dot Uniform Cost Search

Implement UCS graph search

Question 4 (2 points): Finding a Fixed Food Dot A* search

Implement A* graph search

Question 5 (2 points): Finding a Fixed Food Dot Greedy Best First search (GBFS)

Implement Greedy Best First graph search

***** Note that, For each algorithm; calculate the searching time , Explored node and path on the figure .**