

EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the [Privacy Policy](#).



[Course](#) > [Week 3](#) > [Project...](#) > [p1_sea...](#)

p1_search_glossary

Object Glossary

Here's a glossary of the key objects in the code base related to search problems, for your reference:

`SearchProblem` (`search.py`)

A `SearchProblem` is an abstract object that represents the state space, successor function, costs, and goal state of a problem. You will interact with any `SearchProblem` only through the methods defined at the top of `search.py`

`PositionSearchProblem` (`searchAgents.py`)

A specific type of `SearchProblem` that you will be working with --- it corresponds to searching for a single pellet in a maze.

`CornersProblem` (`searchAgents.py`)

A specific type of `SearchProblem` that you will define --- it corresponds to searching for a path through all four corners of a maze.

`FoodSearchProblem` (`searchAgents.py`)

A specific type of `SearchProblem` that you will be working with --- it corresponds to searching for a way to eat all the pellets in a maze.

Search Function

A search function is a function which takes an instance of `SearchProblem` as a parameter, runs some algorithm, and returns a sequence of actions that lead to a goal. Example of

search functions are `depthFirstSearch` and `breadthFirstSearch`, which you have to write. You are provided `tinyMazeSearch` which is a very bad search function that only works correctly on `tinyMaze`.

SearchAgent

`SearchAgent` is a class which implements an Agent (an object that interacts with the world) and does its planning through a search function. The `SearchAgent` first uses the search function provided to make a plan of actions to take to reach the goal state, and then executes the actions one at a time.

© All Rights Reserved