

```

import problemConfiguration as pr
import graphicsUtils as g
# search algorithm, given a problem p and search type
def search(p, searchType="depth"):
    ## *** Your CODE is here ***
    frontier = [] # frontier is an empty list
    ## append start state to the frontier.
    explored = [] # explored list
    statesExplored = 0 # number of nodes explored
    while frontier != []:
        ## get a state from the frontier:
        ## check if the current position is in the explored, if so; skip that state
        ## check if the current position of pacman is the goal state, if so return
the action list and number of nodes explored
        ## append the current position of pacman into explored list
        ## add 1 to the number of states explored
        ## get the legal action:
        ## loop over all legal actions; use for loop
            ## for every action, compute the next state by calling successor
function
            ## check if the new position of pacman after applying the action is
already in explored list, if so, skip that action
            ## append or insert the new state to the frontier.
p = pr.readMaze("maze2.txt")
g.init_graphics(1200, 1200) # initialize window to 900 x 500
g.draw(p) # draw problem p
pathList, nodesExplored = search(p, "breadth") # select the strategy search: depth
or breadth
print(pathList)
print('number of states explored is: ', statesExplored)
if pathList != []:
    g.drawPath(p, pathList) # run the path list and move pacman

```