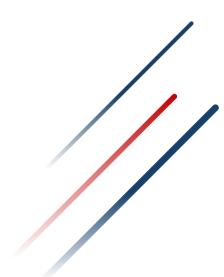


# 人工智能实验

2023春



# 各班课程表



◆ 计算机5-8班上课时间

◆ 计算机9-12班上课时间

7周	星期5	5-6节
8周	星期5	3-4节
9周	星期3	7-8节
10周	星期5	5-6节
11周	星期5	5-6节

7周	星期5	7-8节
8周	星期1	7-8节
9周	星期2	7-8节
10周	星期2	7-8节
11周	星期4	7-8节

◆ 地点: T2608

◆ 地点: T2608

人工智能实验换班登记表 扫一扫二维码打升或分享给好友



\*如有课程冲突需要调整的请填写登记表

# 各班课程表



◆ 计算机1-4班上课时间

◆ 计算机13-16班上课时间

7周	星期2	9-10节
8周	星期2	9-10节
9周	星期3	7-8节
10周	星期3	7-8节
11周	星期5	9-10节

7周	星期1	7-8节
8周	星期1	9-10节
9周	星期4	7-8节
10周	星期4	7-8节
11周	星期4	7-8节

◆ 地点: T2507

◆ 地点: T2210

人工智能实验换班登记表 扫一扫二维码打开或分享给好友



\*如有课程冲突需要调整的请填写登记表

◆ 实验课程共10个学时、2个实验项目,总成绩为30分

实验内容	分值
实验1:搜索算法	15分(报告、代码、考勤等)
实验2: 数字识别	15分(报告、代码、考勤等) +附加题5分

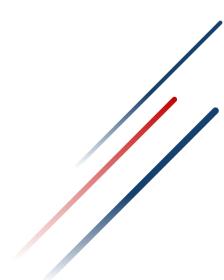
- ◆ 实验一每位同学单独完成;实验二分组完成,每组2-3人
- ◆ 实验课第7~11周,最后一次课为附加题答辩时间



# 人工智能

# 实验1-搜索策略pacman

2023春





pacman是加州大学伯克利分校开源的人工智能实验项目,实验的初衷是在有趣的可视化游戏界面中加入AI策略。

实验地址如下: https://inst.eecs.berkeley.edu/~cs188/sp22/project1/



## 实验目的和内容



> 实验目的

通过pacman实验,加深对各种搜索算法的理解

> 实验内容

在框架代码基础上完成一系列算法,解决8个搜索问题

# 实验内容



# 要求使用但不限于课程讲的各种搜索算法,编写一系列吃豆人程序,解决以下8个搜索问题:

问题1: 应用深度优先算法找到一个特定位置的豆子

问题2:应用宽度优先算法找到一个特定位置的豆子

问题3:应用代价一致算法找到一个特定位置的豆子

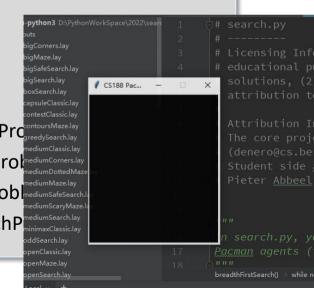
问题4:应用A\*算法找到一个特定位置的豆子

问题5:找到所有的角落——基于BFS的角落问题(CornersProgreedySearchla

问题6:找到所有的角落——基于A\*的角落问题(CornersProb

问题7: 吃掉所有的豆子——食物搜索问题(FoodSearchProblemediumSafeSearch

问题8:次最优搜索——任意食物搜索问题(AnyFoodSearchP



# 实验内容



# 要求使用但不限于课程讲的各种搜索算法,编写一系列吃豆人程序,解决以下8个搜索问题: 要求使用统一的搜索框架

问题1: 应用深度优先算法找到一个特定位置的豆子

问题2:应用宽度优先算法找到一个特定位置的豆子

问题3: 应用代价一致算法找到一个特定位置的豆子Graph Search Pseudo-Code

问题4:应用A\* 算法找到一个特定位置的豆子

问题5: 找到所有的角落——基于BFS的角落问题

问题6:找到所有的角落——基于A\*的角落问题

问题7: 吃掉所有的豆子——食物搜索问题(Food

问题8:次最优搜索——任意食物搜索问题(AnyF

```
function GRAPH-SEARCH(problem, fringe) return a solution, or failure

closed ← an empty set

fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)

loop do

if fringe is empty then return failure

node ← REMOVE-FRONT(fringe)

if GOAL-TEST(problem, STATE[node]) then return node

if STATE[node] is not in closed then

add STATE[node] to closed

for child-node in EXPAND(STATE[node], problem) do

fringe ← INSERT(child-node, fringe)

end

end
```

# 实验内容



# 要求使用但不限于课程讲的各种搜索算法,编写一系列吃豆人程序,解决以下8个搜索问题:

问题1: 应用深度优先算法找到一个特定位置的豆子 应用宽度优先算法找到一个特定位置的豆子 应用代价一致算法找到一个特定位置的豆子 应用A\* 算法找到一个特定位置的豆子 问题4: 找到所有的角落——基于BFS的角落问题(CornersF 找到所有的角落——基于A\*的角落问题(CornersPi 吃掉所有的豆子——食物搜索问题(FoodSearchPro mediumScaryMaze.lay mediumSearch.lay 问题8: 次最优搜索——任意食物搜索问题(AnyFoodSearcl minimaxClassic.lav a oddSearch.lav # openClassic.lay d openSearch.lay

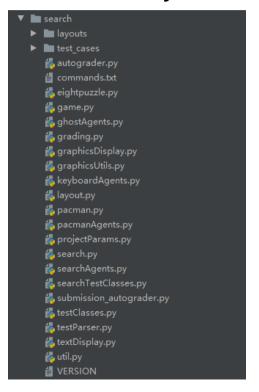


# 要求使用但不限于课程讲的各种搜索算法,编写一系列吃豆人程序,解决以下8个搜索问题:

问题1: 应用深度优先算法找到一个特定位置的豆子 问题2: 应用宽度优先算法找到一个特定位置的豆子 应用代价一致算法找到一个特定 🚜 search.py 应用A\* 算法找到一个特定位置的 a eightpuzzle.py ame.py ghostAgents.py arading.pv 问题5: 找到所有的角落——基于BFS的角 🚜 graphicsDisplay.py 🐍 graphicsUtils.py 找到所有的角落——基于A\*的角落 & keyboardAgents.py alayout.py 🐍 pacman.py Language pacman Agents.py 吃掉所有的豆子——食物搜索问题 self.searchFunction = lambda prob: sea projectParams.py & search.py search.py.bak 问题8:次最优搜索——任意食物搜索问题 a searchAgents.py searchAgents.pv.bak def foodHeuristic(state, problem): searchTestClasses.py 🐔 submission autograder.py testClasses.py testParser.py textDisplay.py



#### 整个项目使用Python开发



#### 需要阅读代码的文件

文件	主要功能
pacman.py	吃豆人游戏的主程序
game.py	吃豆人游戏的运行逻辑
util.py	提供一些可用的数据结构,如Stack、 Queue、PriorityQueue

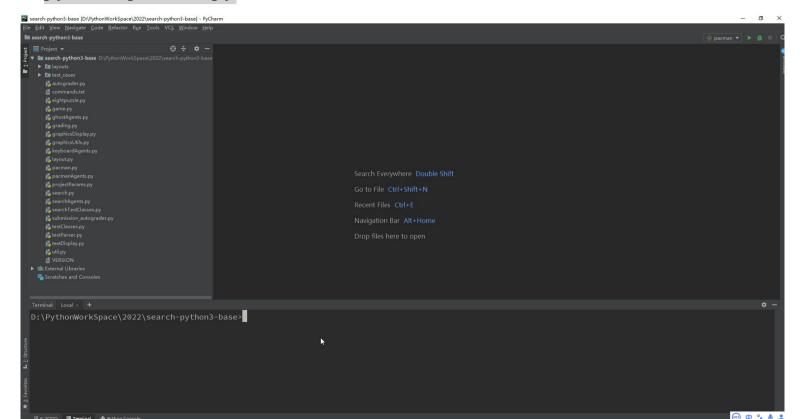
#### 需要完善代码的文件

文件	主要功能
search.py	待实现的搜索算法
searchAgents.py	待实现的智能体和相关Problem

# 运行和测试代码

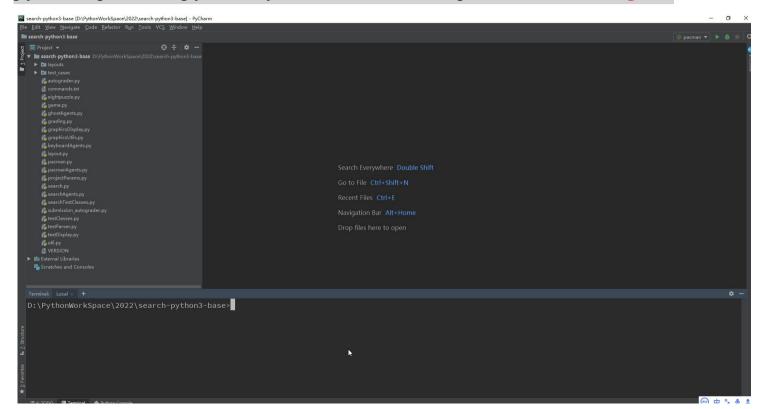


#### python pacman. py



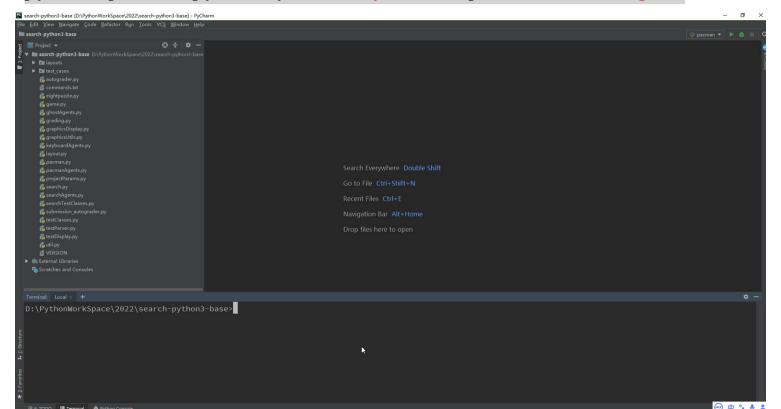
#### 运行和测试代码

#### python pacman.py --layout testMaze --pacman GoWestAgent



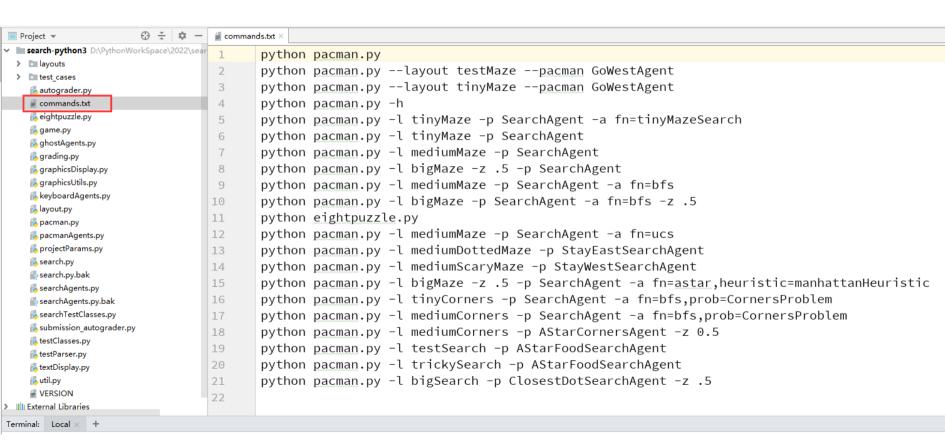
#### 运行和测试代码

#### python pacman.py --layout tinyMaze --pacman GoWestAgent



#### 测试命令





# 常用的命令参数



设置图形窗口的缩放比例 指定使用的agent类型。默认是1,比1大则是放 大,比1则是缩小

python pacman.py -I bigMaze -p SearchAgent -a fn=bfs -z .5

search-python3-base D:\Pytho ▼ layouts bigCorners.lay 間 bigMaze.lay # bigSafeSearch.lay bigSearch.lay BoxSearch.lay apsuleClassic.lay d contestClassic.lay d contoursMaze.lay greedySearch.lay # mediumClassic.lay # mediumCorners.lay # mediumDottedMaze.lay mediumMaze.lay mediumSafeSearch.lay # mediumScaryMaze.lay mediumSearch.lay

地图选项参数,从layouts/目录下加载指定类型地图

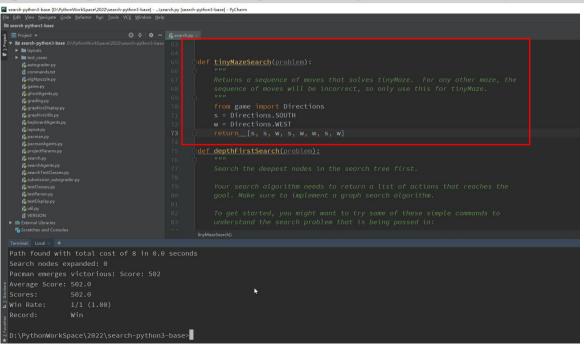
传递给agent的参数值。以 字符串的形式,如果有多个 取值以逗号分隔

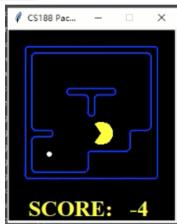


#### 问题1:应用深度优先算法找到一个特定位置的豆子

首先,运行以下命令测试SearchAgent是不是正常工作:

python pacman.py -1 tinyMaze -p SearchAgent -a fn=tinyMazeSearch







问题1:应用深度优先算法找到一个特定位置的豆子

python pacman.py -1 tinyMaze -p SearchAgent -a fn=<mark>dfs</mark>

```
D:\PythonWorkSpace\2022\search-python3-base>python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
[SearchAgent] using function dfs
[SearchAgent] using problem type PositionSearchProblem

*** Method not implemented: depthFirstSearch at line 90 of D:\PythonWorkSpace\2022\search-python3-base\search.py
```

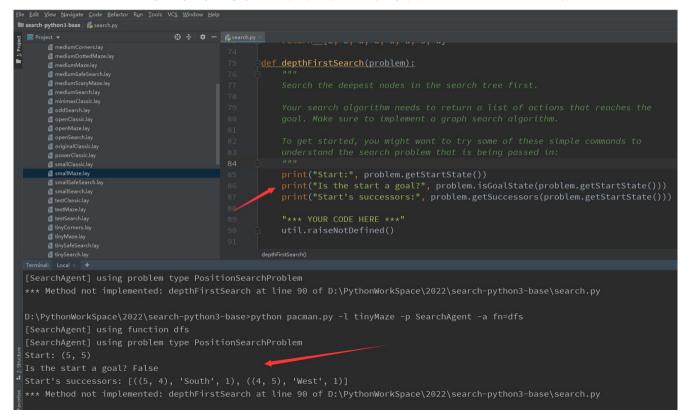


#### 问题1:应用深度优先算法找到一个特定位置的豆子

```
w = Directions.WEST
def depthFirstSearch(problem):
    "*** YOUR CODE HERE ***"
    util.raiseNotDefined()
def breadthFirstSearch(problem):
    "*** YOUR CODE HERE ***"
    util.raiseNotDefined()
def uniformCostSearch(problem):
    "+++ VOLID CODE MEDE +++"
```

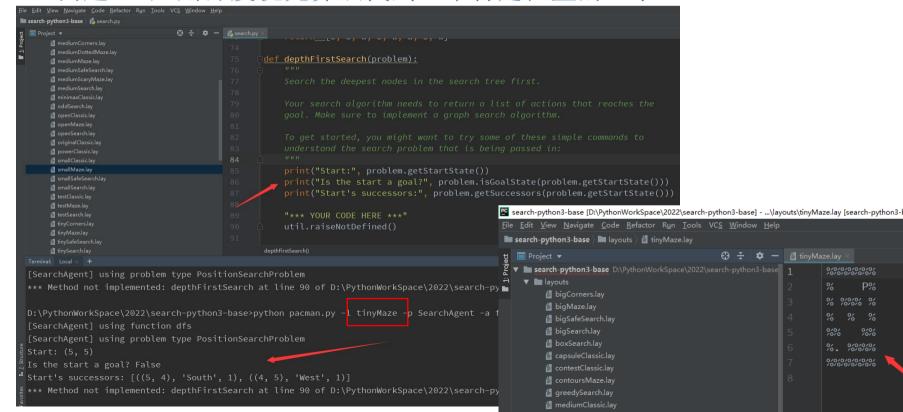


#### 问题1:应用深度优先算法找到一个特定位置的豆子





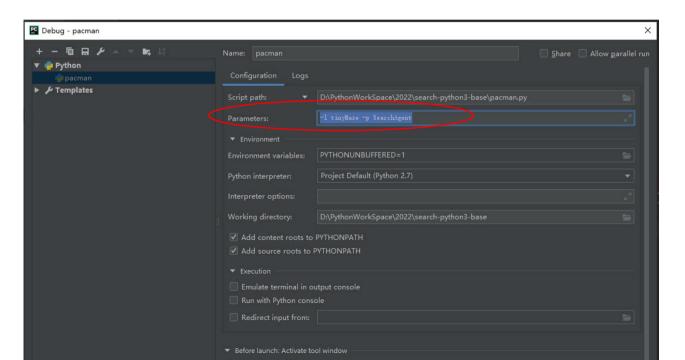
#### 问题1:应用深度优先算法找到一个特定位置的豆子





#### 问题1:应用深度优先算法找到一个特定位置的豆子

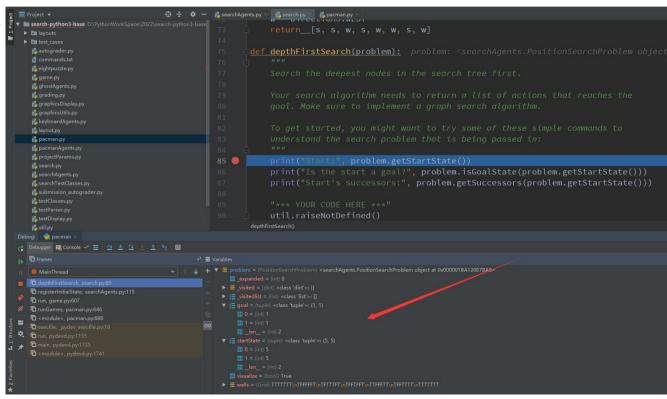
▶ 用Debug跟踪执行





#### 问题1:应用深度优先算法找到一个特定位置的豆子

▶ 用Debug跟踪执行





#### 问题1:应用深度优先算法找到一个特定位置的豆子

#### > 测试代码

• 补充完depthFirstSearch方法后,先用以下命令进行测试:

python pacman.py -1 tinyMaze -p SearchAgent

nython pacman.py -1 modiumMaze -p SearchAgent

python pacman.py -1 mediumMaze -p SearchAgent python pacman.py -1 bigMaze -z .5 -p SearchAgent

• 以上3个测试都通过并不代表完全正确,需要运行以下命令对问题1 进行更完整的测试:

python autograder.py -q q1

-q选项是指定对应的问题



#### 运行python autograder.py

```
D:\PythonWorkSpace\2022\search-python3>python autograder.py
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for al
  import imp
Starting on 3-29 at 17:10:40
Question q1
*** PASS: test cases\q1\graph backtrack.test
       solution:
                     ['1:A->C', '0:C->G']
       expanded states: ['A', 'D', 'C']
*** PASS: test cases\q1\graph bfs vs dfs.test
       solution:
       expanded states: ['A', 'D']
*** PASS: test cases\q1\graph infinite.test
       solution:
                     ['0:A->B', '1:B->C', '1:C->G']
       expanded states:
                        ['A', 'B', 'C']
*** PASS: test_cases\q1\graph_manypaths.test
       solution:
                    ['2:A->B2', '0:B2->C', '0:C->D', '2:D->E2', '0:E2->F', '0:F->G']
       expanded states:
*** PASS: test_cases\q1\pacman_1.test
                              mediumMaze
```



#### 运行python autograder.py

```
Finished at 17:11:00
 Provisional grades
 Question q1: 3/3
 Question q2: 3/3
 Question q4: 3/3
 Question q6: 3/3
 Question q7: 5/4
 Question q8: 3/3
 Total: 26/25
 Your grades are NOT yet registered. To register your grades, make sure
to follow your instructor's guidelines to receive credit on your project.
```



#### Q6: 找到所有的角落——基于A\*的角落问题(构建启发函数)

python pacman.py -1 mediumCorners -p AStarCornersAgent -z 0.5

Number of nodes expanded	Grade
more than 2000	0/3
at most 2000	1/3
at most 1600	2/3
at most 1200	3/3

不同的启发函数效率不同,<mark>扩展的节点数</mark>不同,autograder.py会根据不同的节点数给分



#### Q7: 吃掉所有的豆子——食物搜索问题 (A\*算法,构造启发函数)

python pacman.py -1 trickySearch -p AStarFoodSearchAgent

Number of nodes expanded	Grade
more than 15000	1/4
at most 15000	2/4
at most 12000	3/4
at most 9000	4/4 (full credit; medium)
at most 7000	5/4 (optional extra credit; hard)

不同的启发函数效率不同,<mark>扩展的节点数</mark>不同,autograder.py会根据不同的节点数给分

## 实验要求



- 独立完成,严禁抄袭;
- 所有操作必须合法(比如不能翻越墙壁);
- ▶ 利用util.py文件中提供的Stack、Queue 和 PriorityQueue 数据结构。这是自动评分系统的兼容性要求。

# 提交说明



- ▶ 提交物:
  - 整个工程 (不要仅提交.py文件)
  - 实验报告 (一定要用模板)
- ➤ 作业提交平台: <a href="http://grader.tery.top:8000/#/courses">http://grader.tery.top:8000/#/courses</a>
- ▶ 截止时间:5月5日

# Python学习



- python基础需自主学习,后续深度学习实验继续使用python完成,版本要求为python3
  - CS188 <u>Python/Autograder Tutorial</u>
  - python官方教程中文版
  - 廖雪峰python教程

> 学习重点: python安装、基本的数据类型、函数、模块等基础



本课程的实验二是分组完成,每组2-3人今明两天内组好队,填下共享文档

5-8班



人工智能实验二分组情况5-8...

扫一扫二维码打开或分享给好友



9-12班



人工智能实验二分组情况9-1...

扫一扫二维码打开或分享给好友



# 华为云代金券



➤ 实验二有个附加题需要用到华为云,如计划做此附加题,请在共享文档填写华为云账号,以便申请代金券





# 同学们 请开始实验吧!