上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

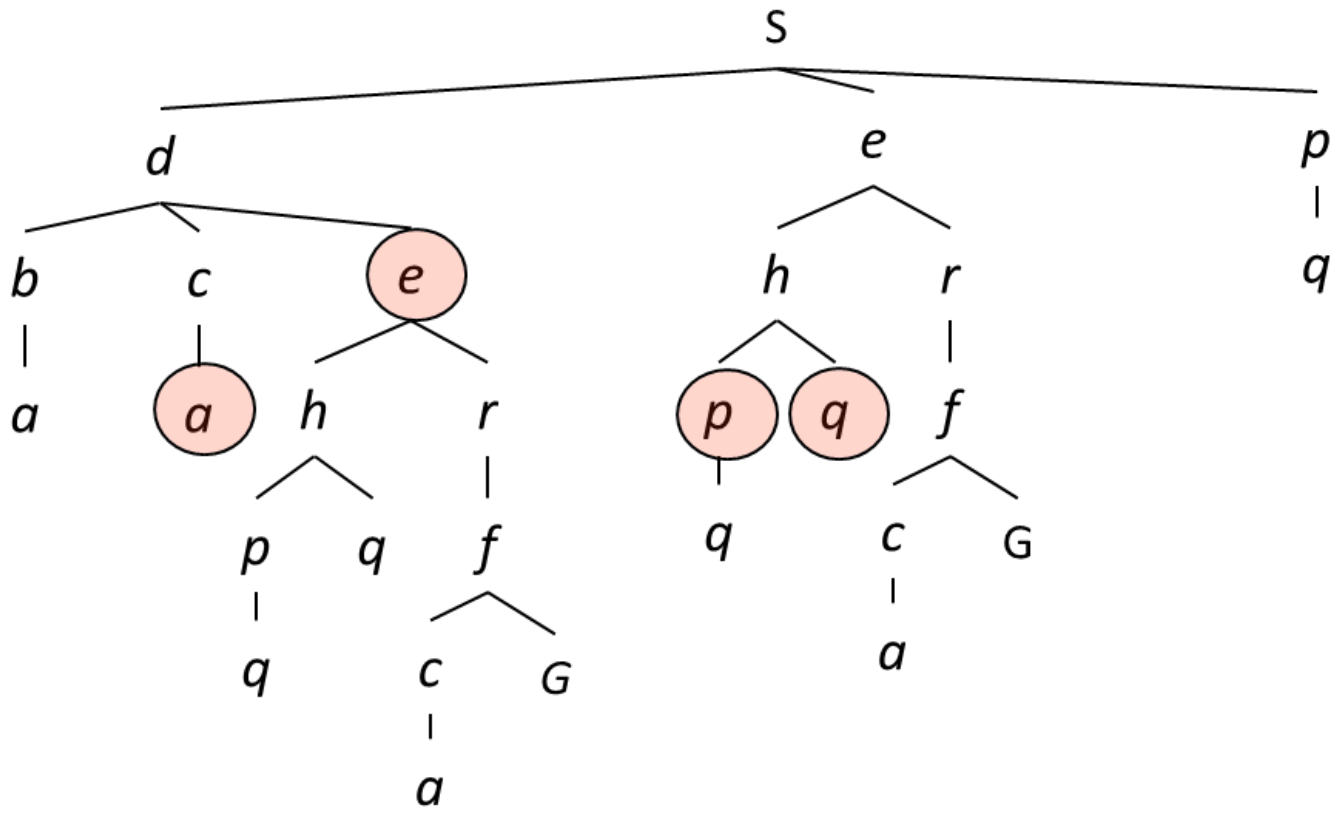
搜索算法实践

马超

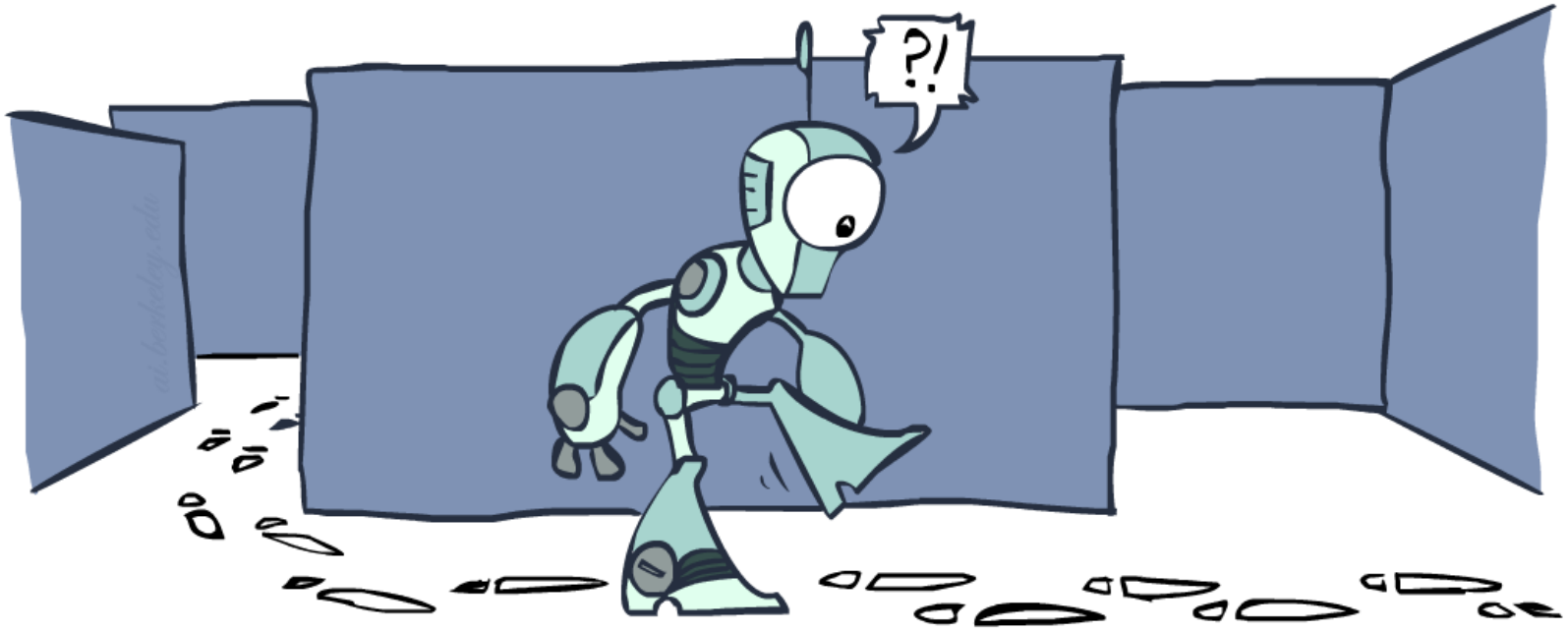
2025年3月15日

搜索算法实践 — 图搜索算法

- 在广度优先搜索中，我们并不需要搜索这些圈内的节点。为什么?



搜索算法实践 — 图搜索算法

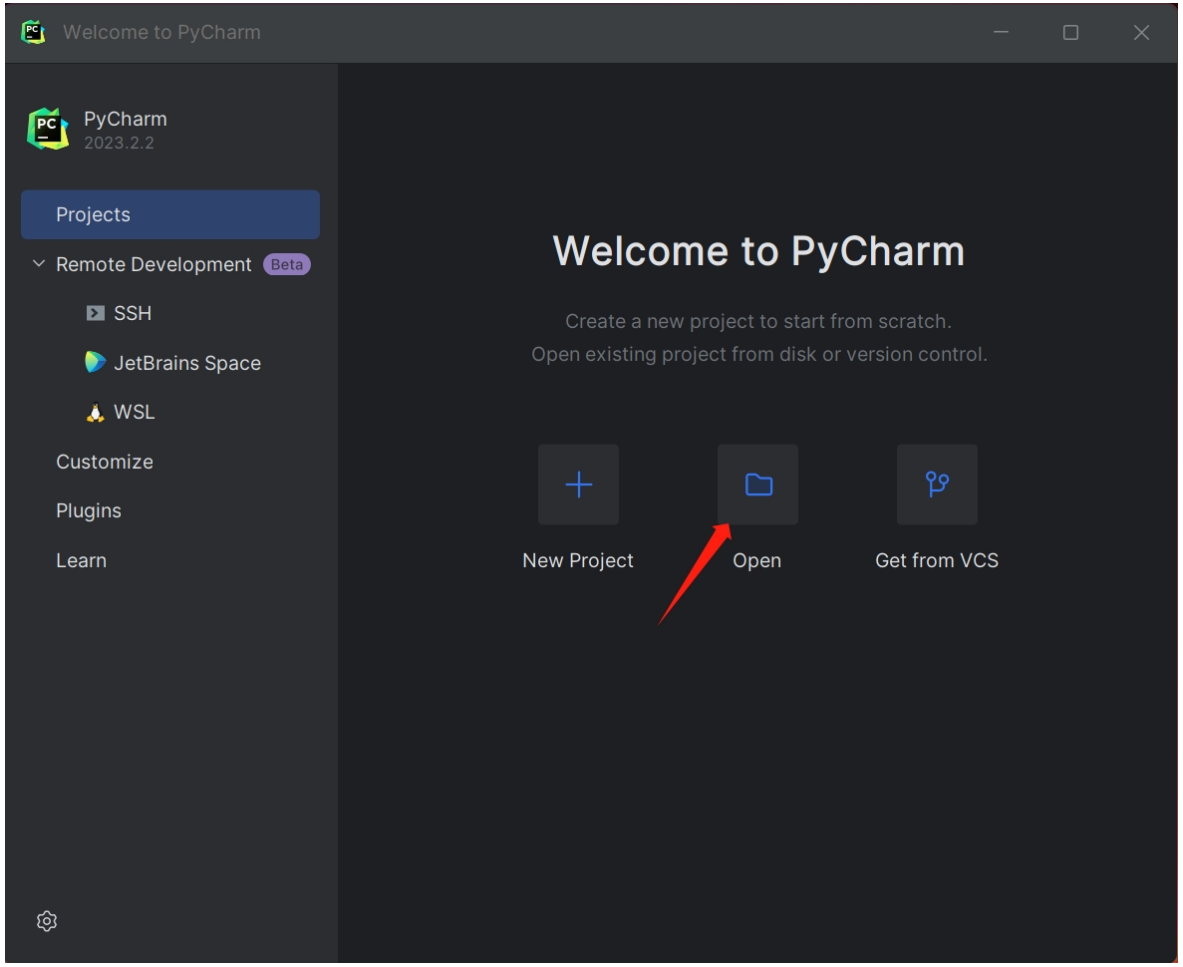


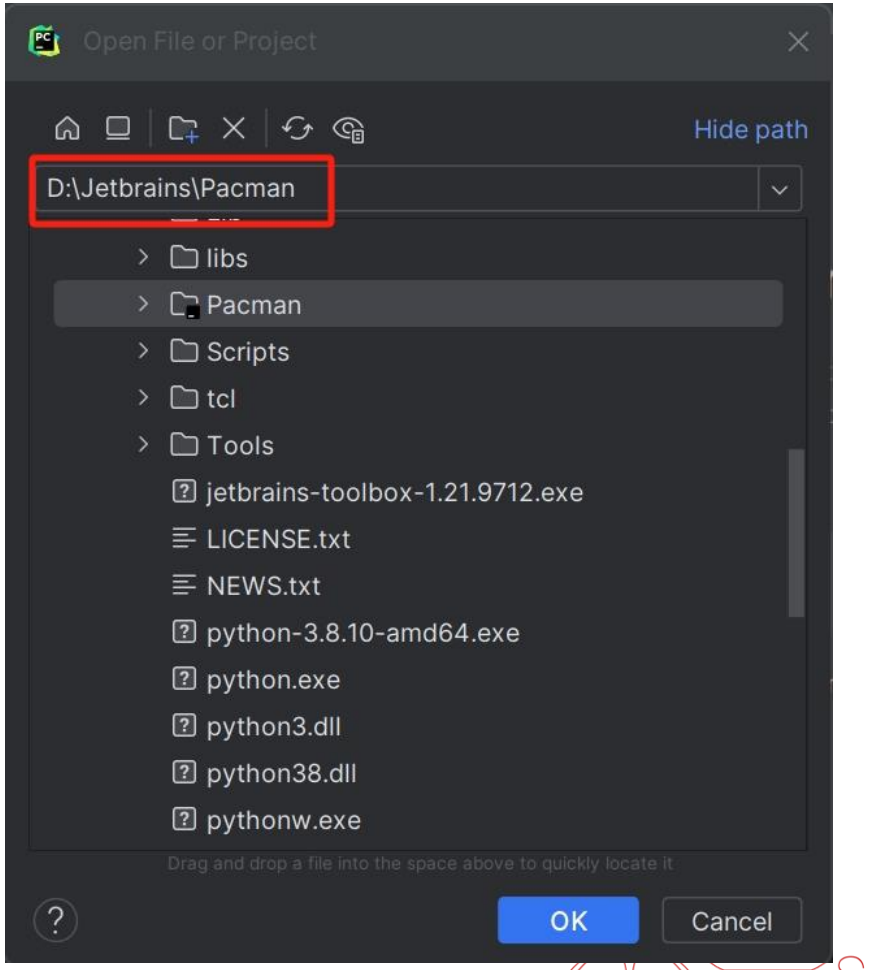
搜索算法实践 — 图搜索算法

- 主要思想：不重复搜索一个节点
- 实现逻辑：
  - 树搜索 + 一个记录已经搜索过节点的集合（closed set)
  - 按照树搜索的顺序逐点搜索，但是在搜索某个结点之前，先判断其是否被搜索过，即是否在closed set中
    - 若被搜索过，则跳过
    - 若未被搜索过，则按照树搜索的方法正常搜索，并把它加入closed set

搜索算法实践 — 搜索算法在吃豆人中的应用

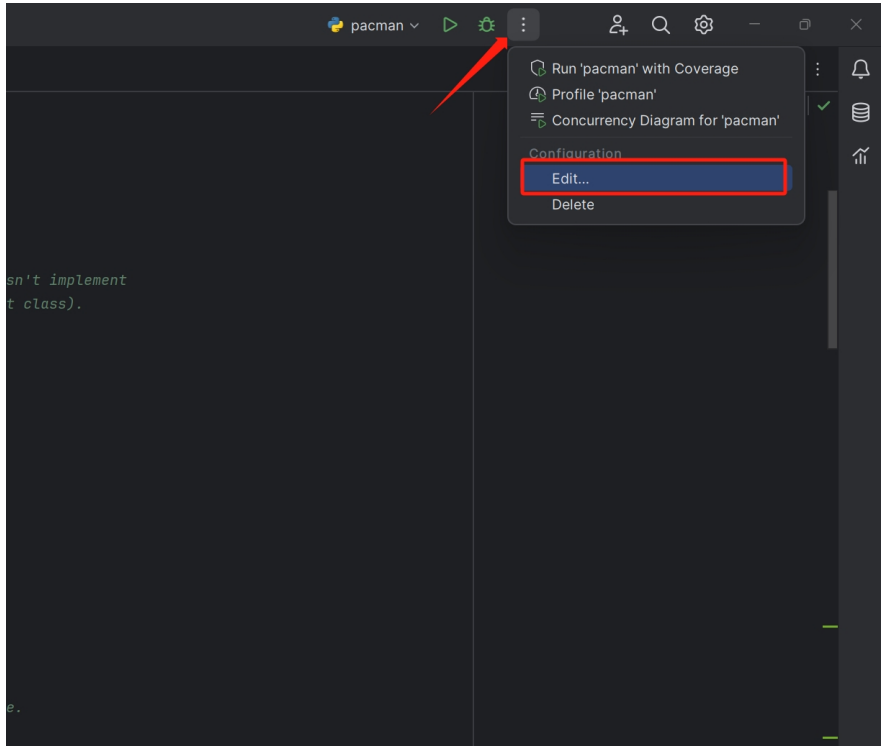
- 下载代码库，解压，放到 D:\Jetbrains\Pacman 路径
- 使用PyCharm打开D:\Jetbrains\Pacman文件夹，并打开pacman.py文件

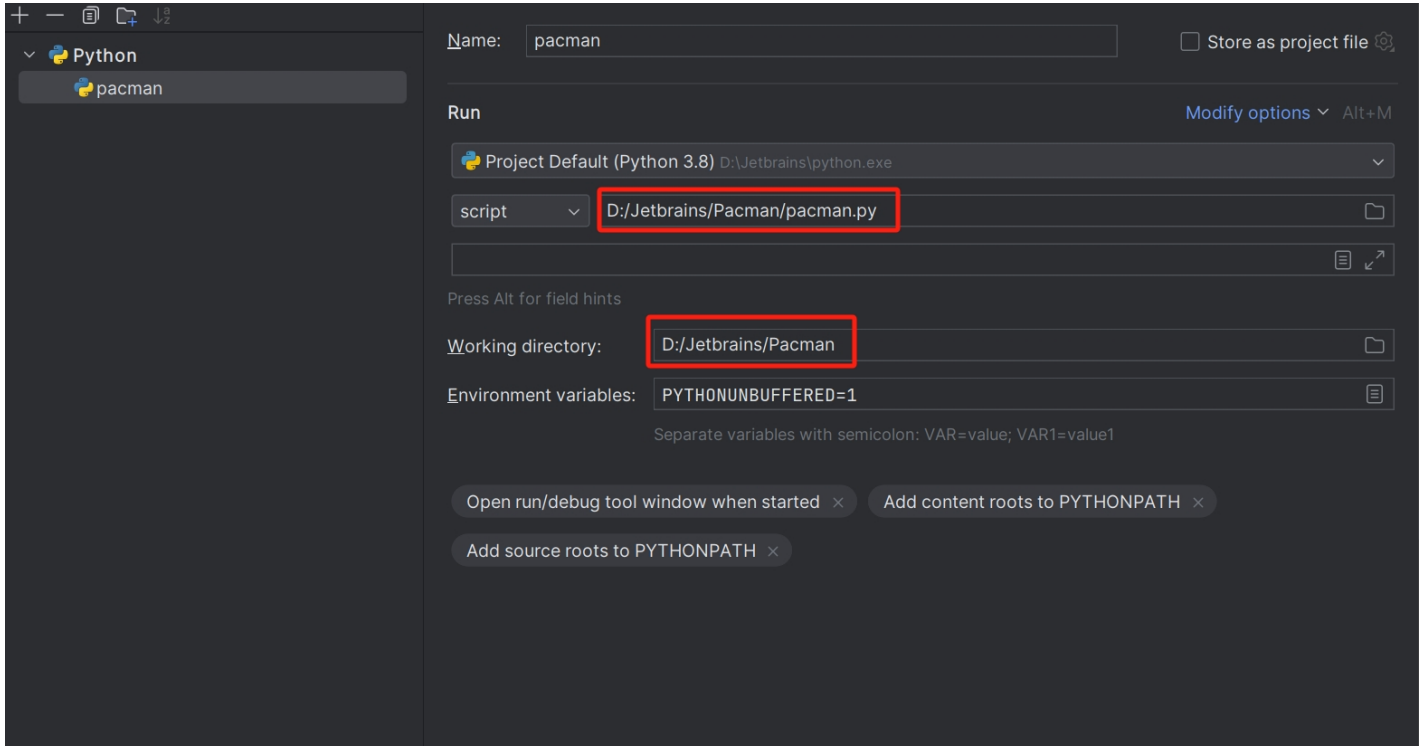




搜索算法实践 — 搜索算法在吃豆人中的应用

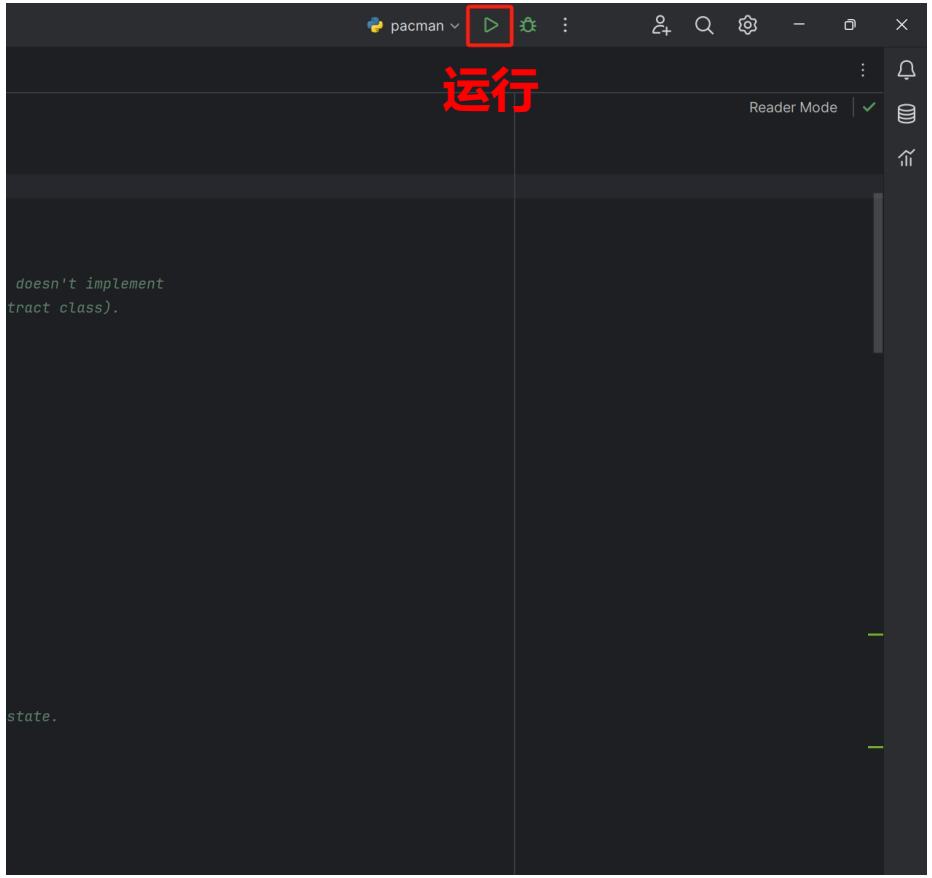
- 点击Run – Edit Configurations – Add new – Python
- 在Script path处写入pacman.py的存储地址
- 在Working directory处写入D:\Jetbrains\Pacman文件夹的地址

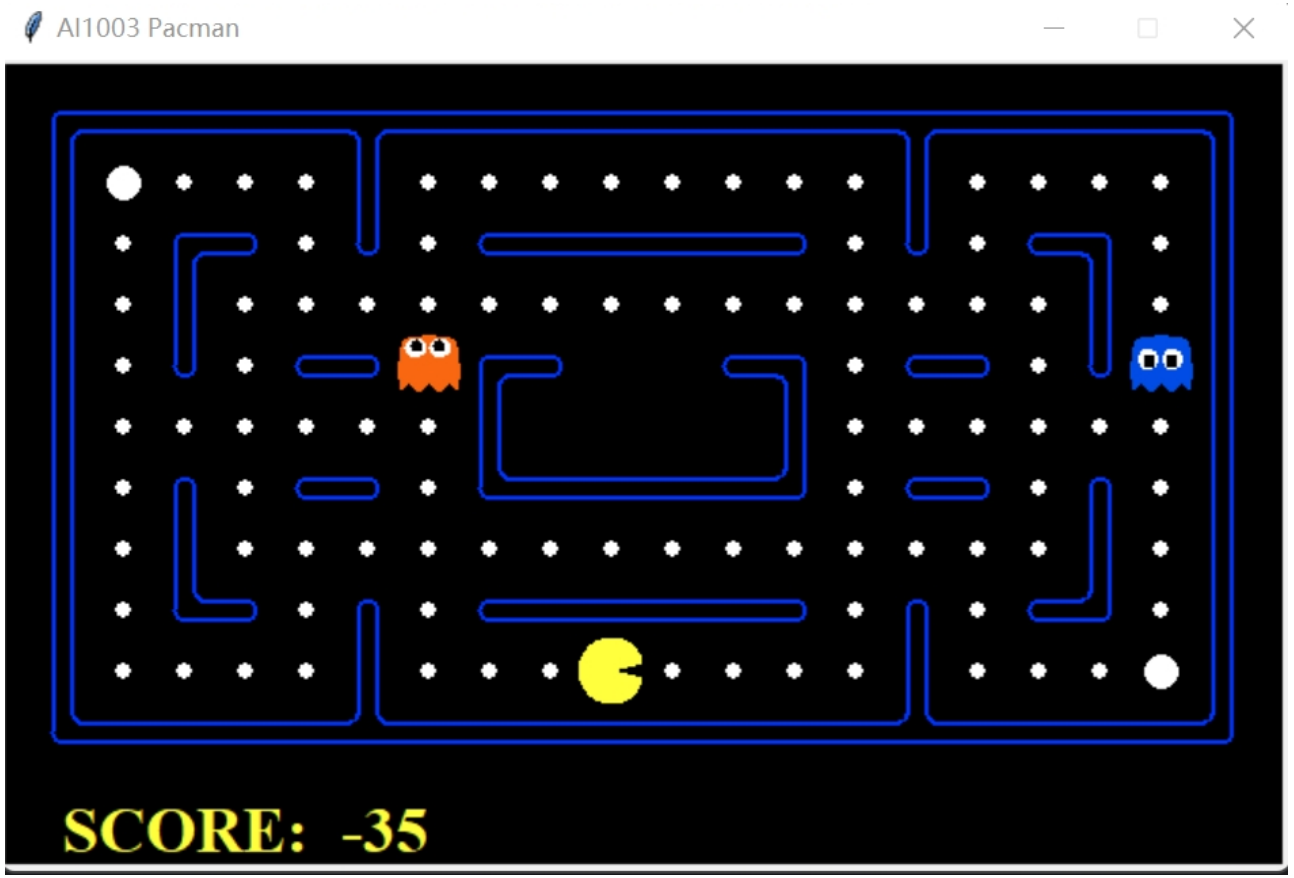




3搜索算法实践 — 搜索算法在吃豆人中的应用

- 点击Apply，然后运行程序，会弹出吃豆人窗口





搜索算法实践 — 搜索算法在吃豆人中的应用

- 1.在代码中TODO部分补全search.py中四种不同搜索算法的代码
  - depthFirstSearch
  - breadthFirstSearch
  - uniformCostSearch

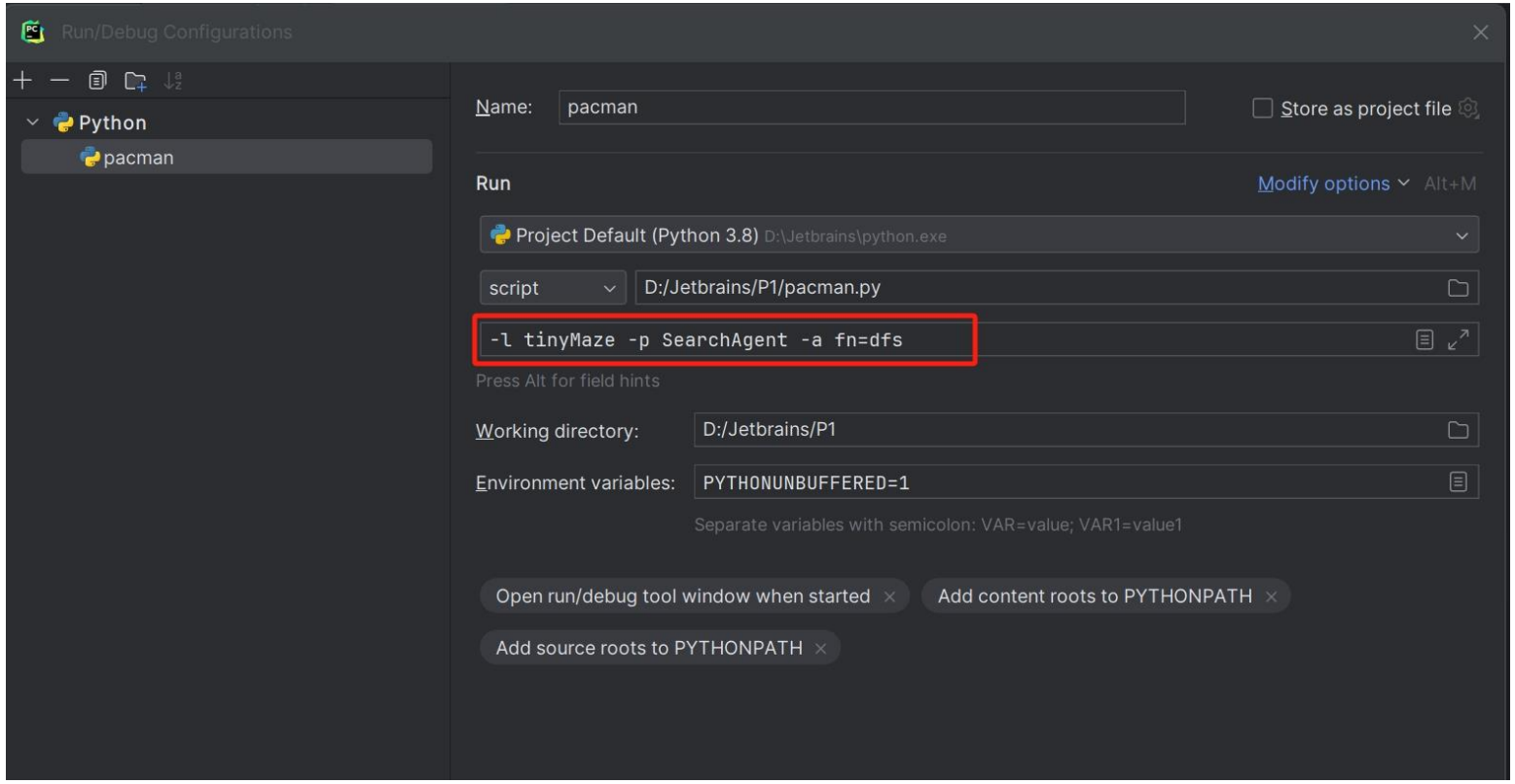
```
def breadthFirstSearch(problem):
    """Search the shallowest nodes in the search tree first."""
    "*** YOUR CODE HERE ***"
    closed = set()
    fringe = _ # TODO util.Stack() OR util.Queue()
    node = {
        'position': problem.getStartState(),
        'path': []
    }
```

搜索算法实践 — 搜索算法在吃豆人中的应用

- 2. 代码测试 (以深度优先为例)

测试方法：点击Run – Edit Configurations

在Parameters处写入-l tinyMaze -p SearchAgent -a fn=dfs，点击apply



搜索算法实践 — 搜索算法在吃豆人中的应用

- 2. 代码测试
  - 深度优先：python pacman.py -l tinyMaze -p SearchAgent -a fn=dfs
  - 广度优先：-a后替换为 fn=bfs
  - 一致代价：-a后替换为 fn=ucs

  - 测试时可以将 –l 之后的tinyMaze换成mediumMaze或者bigMaze来测试不同的地图，在使用bigMaze时可加上：-z 0.5来显示完整的地图，并使用--frameTime 0来加快Pacman的移动速度。

搜索算法实践 — 搜索算法在吃豆人中的应用

- 3. 对比实验
  - 将 –l 后的 tinyMaze 替换为 mediumMaze 和 bigMaze，测试三种搜索算法在三种不同的地图上的搜索表现
  - 要求：记录实验中的搜索到路径的长度“total cost”以及搜索的节点数“nodes expanded”，将结果记录在results.xlsx
  - 注：在测试 bigMaze 时，可加上 -z 0.5 来显示完整的地图

```
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 14
```

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

人工智能研究院
Artificial Intelligence Institute

谢 谢！

饮 水 思 源　　爱 国 荣 校