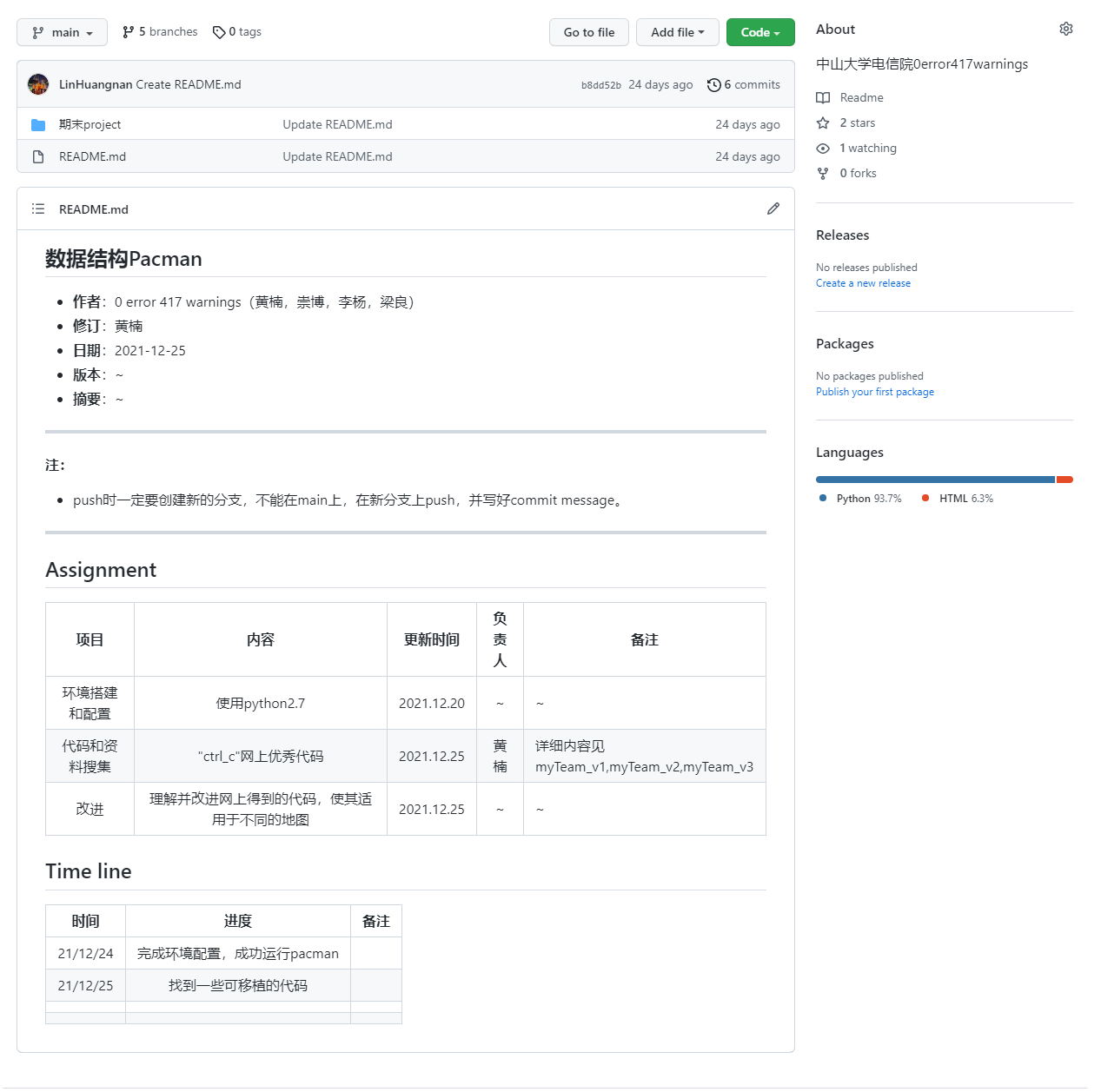
数据结构与算法课程大作业报告

（李杨\* 林黄楠 廖崇博 梁良）

# 一．分工与合作

**1.小组的分工合作主要基于GITHUB**

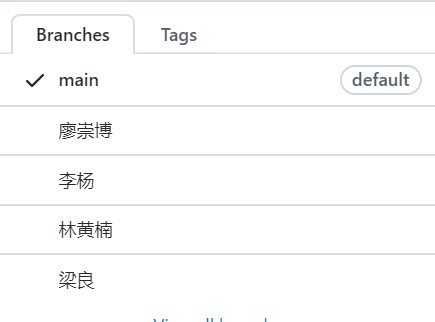
我们在GITHUB上面建立了一个仓库：

我们选择GITHUB进行合作的原因是GITHUB主要可以完成两件事情：

①版本控制

②多人协作开发

这次的吃豆人大作业，项目规模大，难度高，需要团队的四个人合作进行开发，基于此，我们创建了四个分支，小组成员分别在自己名字的分支下面修改和提交自己的代码，最后将每个人的代码进行PK，取其精华，去其糟粕，最后把每个人代码的优点整合起来，改进出最终的算法。



# 

# 二． 经验与教训

**1.得意之处**

**（1）学习了A\*搜索算法**

A\* （A-Star)算法是一种静态路网中求解最短路径最有效的直接搜索方法，也是许多其他问题的常用启发式算法。公式表示为： f\*(n)=g\*(n)+h\*(n),其中， f\*(n) 是从初始状态经由状态n到目标状态的最小代价估计，g\*(n) 是在状态空间中从初始状态到状态n的最小代价，h\*(n) 是从状态n到目标状态的路径的最小估计代价。（对于路径搜索问题，状态就是图中的节点，代价就是距离）

真实h(n)的选取：

保证找到最短路径（最优解的）条件，关键在于估价函数f(n)的选取（或者说h(n)的选取）。

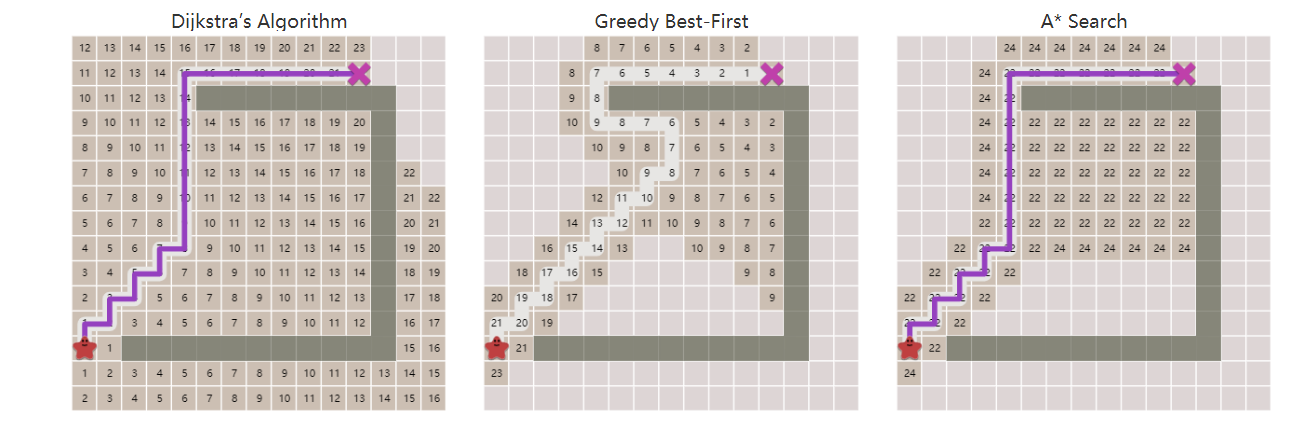
以h(n)表达状态n到目标状态估计的距离，那么h(n)的选取大致有如下三种情况：

① 如果h(n)< h\*(n)，这种情况下，搜索的点数多，搜索范围大，效率低。但能得到最优解。

② 如果h(n)=h\*(n)，此时的搜索效率是最高的。

③ 如果 h(n)>h\*(n)，搜索的点数少，搜索范围小，效率高，但不能保证得到最优解。

下图为3种搜索算法实际效果的比较（图源Introduction to the A\* Algorithm (redblobgames.com)），可以发现，A\*算法可以克服贪心算法的一些缺陷，且与Dijkstra's algorithm算法相比复杂度明显减小



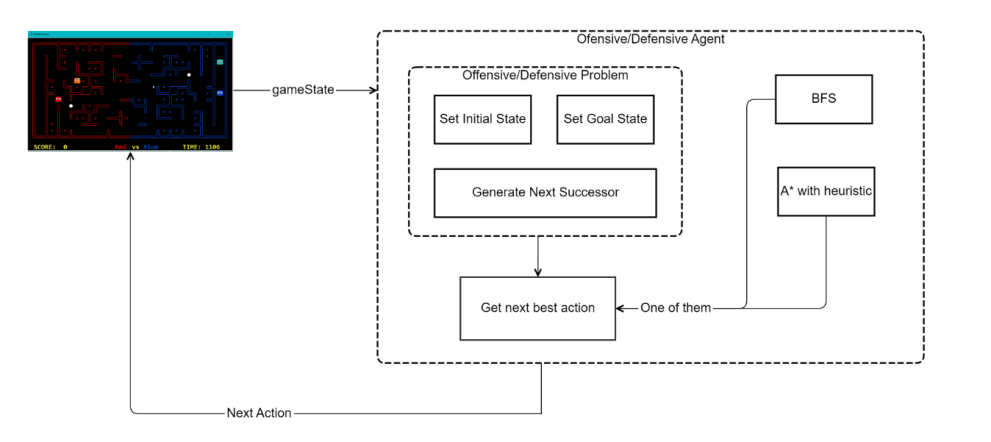
在不同的决策树所判定的情况下选择不同的搜索目标，再采用A\*算法搜索路径，其中，选取了h = （5-x）5其中x为当前点到目标点的曼哈顿距离

注意最优路径的选取与启发函数h(n)相关，宽度优先搜索和Dijkstra算法保证在给定输入图的情况下找到最短路径。如果启发函数永远不大于真实距离，则A\*保证找到最短路径。随着启发式变得更小，A\*变成了Dijkstra算法。当启发函数变得更大时，A\*变成贪婪的最佳优先搜索。

1. **决策树**

决策树简单来说就是一堆if-else的组合。

我们在游戏中使用决策树来决定目标状态，并找到从当前状态到目标状态的最佳路径。采取的动作的计算步骤如下：



我们将我们的两个吃豆人被分为进攻者和防御者

①游戏开始时，我们记录游戏的初始状态和起始位置，这对进攻和防守吃豆人都是一样的。

②两个吃豆人都定义了选择动作（action）的方法,并且我们定义问题（problem）。对于进攻者，我们有offensiveProblem，对于防守者，我们定义了defensiveProblem。定义这些问题是为了将它们传递给BFS和A\*，以产生动作（action）的列表

③进攻者和防守者使用各自的类来定义问题problem，并将其传递给搜索算法（BFS或A\*）

④我们找到使用搜索方法的最佳方案。通过上面的步骤我们会得到一个动作列表，我们选择第一个动作作为吃豆人的将采取的下一个动作。这个动作会导致一个新的游戏状态，然后重复上述步骤，才能在新的游戏状态下找到下一个动作。

1. **不同的情况以及采取的策略**

我们首先将食物定义为危险食物或安全食物，危险食物即只有一条路可以到达食物位置，因此我们的pacman去吃时很可能会陷入危险的情况，而安全食物则是有两条路可以到达食物位置，因此我们的pacman去吃时有较大可能逃生。然后我们根据这两种食物状态的区分用if-else决策树来决定攻击者和防守者的行动方式。

①对于进攻者：

当场上没有安全的食物，场上还有胶囊，对手处于恐吓剩下的时间不多时，pacman将会进入搜索胶囊的状态，为之后食用危险的食物做准备；当pacman身上没有携带食物且场上存在安全的食物时，pacman将进入食用安全的食物的状态；当pacman身上没有携带食物且场上没有安全的食物时，pacman将不会进行安全食物和危险食物的区分，直接去搜索食物；当鬼出现且与pacman相距不远时，pacman将会进入逃生状态；当大力丸的持续时间足够时，pacman将会进入搜索危险食物到达状态；当场上剩余食物小于3或游戏时间只是稍多于回家时间或身上携带食物数量大于15时，pacman将进入回家的状态；当以上的状态都没有进入时，pacman将直接进入搜索食物的状态，不区分食物的安全与否。

②对于防守者：

当已知的入侵者数量为零时：我们的防守者将会尝试去搜索食物，当携带食物小于3，场上食物不为零，自身状态相对安全的情况下，将会进入搜索食物的状态，否则，将会进入回家的状态；

当己方有食物被吃时但入侵者位置未知时，防守者将会前往上次被吃的食物位置，以期望能找到入侵者，当入侵者位置已知时，防守者将会去追逐入侵者；

当以上情况都没有发生时，我们将使用feature和weight来决定移动，防守者将表现为在边界上下行走知道入侵者位置或最后被吃的食物位置已知。

**2.经验与教训**

我们发现了我们的代码有一些普遍问题并且总结了一些比较好的想法：

①当我们的防守者长时间处于被恐吓的状态时，我们的攻击者并不会换回防守者来协助防御，这就导致当对面吃到我们的大力丸时我们的防守能力下降，甚至完全失去防守能力。

②出现僵局。出现僵局的原因，是由于与对方防御者的距离太远，在动作选择时，距离因素在阈值上下浮动，为了改进这个情形，我们将代码修改了好几次，虽然有所改进，但是还是无法完全避免。

③当我们的攻击者吃到一定数量的豆子时，它会有两种选择逃脱，一种是寻找大力丸，另一种是寻找最近的回家的路，并且当吃到大力丸时，我们的攻击者会优先选取危险豆子来吃，来弥补正常状态下不敢吃危险豆子的缺点。

# 三． 建议与设想

**1.对本次实验的建议**

在最终的比赛过程中，由于班级人数较多且需要循环比赛故实际比赛的时间较长，除此之外在最终测试前我们不了解自己代码的实际表现。故我们希望能够增加在线的对战平台以提高对战的效率和进行赛前的改进。

**2.寄语**

数据结构结构是计算机课程类的核心，我们在学习数据结构课程过程中我们要抓住核心的、抽象的数据结构并了解实现他们的基本方法。同时由于计算机相关知识十分庞杂、更新迭代速度十分迅速，我们要扩大自己的视野了解前沿的相关知识。

同时数据结构本身是抽象的，我们要了解实际工程中数据结构的应用，并且要主动学习基础语法以外的编程知识，多学习与实际工程相关的知识，能够做出除控制台应用以外的程序。

在实际完成项目中，代码的版本控制、命名的规范、注释的添加、文件的整理都对团队合作有很大的帮助，我们要学习多人合作来完成困难的复杂的目标。

1. **设想**

在本次Pacman实验过程智能体的决策是多种多样的，我们设想能够将最优化、博弈论甚至是机器学习相关的知识、算法结合到实验中以得到更优的决策。

1. **教学建议：**

平衡教学时间，适当减少对python基础的讲解而提高对一些数据结构的教学时间。

另外希望能够提高OJ平台的稳定性，在完成作业后可添加一定的讲解。

Pacman的说明可以添加一定命令行的讲解，降低上手pacman的难度。