**2021年春季学期**

**数据结构课程设计A3题实验报告**

郑修远1,\* 贾燕鹏2 陈敏杰3郭军豪4

1计算机科学与技术学院2019级31班

2计算机科学与技术学院2019级31班

3计算机科学与技术学院2019级31班

4计算机科学与技术学院2019级31班

\*表示队长

摘要：使用了UCT算法，即为UCB算法处理博弈树的思想：通过多次模拟的结果，寻找到概率最高的那一个节点。将主要精力用在这一个节点上，避免不必要的浪费，即利用（Exploitation）。但是，也要照顾到那些被“冷落”的节点，避免失去机会，即探索（Exploration）。使用UCB公式来决定利用与探索的比重。

结果排名7/28，分数55/75。

关键字：UCB算法 蒙特卡洛树

[1 分工与合作 1](#_Toc69548504)

[2 算法思想 2](#_Toc69548505)

[2.1 总体思路 2](#_Toc69548506)

[2.2 所用方法的特别、新颖或创新之处 2](#_Toc69548507)

[2.3 算法流程图 2](#_Toc69548508)

[2.4 算法运行时间复杂度分析 2](#_Toc69548509)

[3 程序代码说明 2](#_Toc69548510)

[3.1 数据结构说明 2](#_Toc69548511)

[3.2 函数说明 2](#_Toc69548512)

[3.3 程序限制 2](#_Toc69548513)

[4 实验结果 3](#_Toc69548514)

[4.1 测试方法与数据 3](#_Toc69548515)

[4.2 结果分析 3](#_Toc69548516)

[4.3 经典战局 3](#_Toc69548517)

[5 总结 3](#_Toc69548518)

[6 参考文献 3](#_Toc69548519)

# 分工与合作

郑修远：进行框架搭建，算法代码实现

贾燕鹏：算法代码实现，程序调试与bug处理

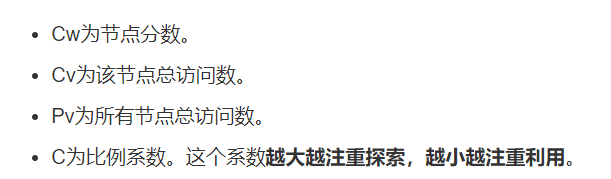
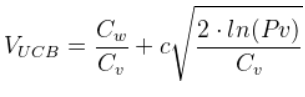
陈敏杰：程序调试处理bug，测评并进行参数的设定调整

郭军豪：测评并进行参数的设定调整，撰写试验报告。

# 算法思想

## 总体思路

总体上采用UCT算法。通过建立蒙特卡洛树，并采用UCB公式来进行判断，总重得出可前进各个方向的权值来判断优解。UCB算法利用下图公式来求出要选择的结点，并通过参数c的调整来决定利用和探索的比重，从而得出有限时间内算力得到最好利用的程序，从而建立bot。程序最早来自nogo的uct程序框架，经过讨论与尝试得出了最适合snake的uct代码实现，并经过一步步修改调整形成最终bot。



## 所用方法的特别、新颖或创新之处

（矩阵：

一层模拟敌我3\*3共九个结点：

## 算法流程图

（我手太拉画不出

## 算法运行时间复杂度分析

（

# 程序代码说明

## 数据结构说明

总体建立了蒙特卡洛树。

树的节点包括分数，模拟次数，子节点个数，方向，有效方向，父亲节点，子节点数组等。

## 函数说明

bool whetherGrow() ：本回合是否生长

void deleteEnd(int id) ：删除蛇尾（编号为id蛇）

void move(int id, int dir) ：编号为id的蛇朝向dir方向移动一步

bool validDirection(int id, int k) ：判断当前移动方向的下一格是否合法，传参为蛇与方向

int getValidDir(int\* dir, int id) ： 取得合法移动方向

void backup(pn p, int res) ：回退，传结点指针和结果分数

int randMove()： 随机移动，返回方向值

pn bestchild(pn p) ：寻找最优子节点

pn MTCS()： 建立蒙特卡洛树返回结点指针

## 程序限制

无

# 实验结果

## 测试方法与数据

总体通过实际对战来对算法进行测试。

早期通过出错和超时的错误来寻找bug，并辅助调试。后期通过对战的胜率与debug字段的运用来调整得到合适的参数值。

## 结果分析

UCB公式起到了作用，测试过程中通过c值的调整，程序强度得到了优化。运行时间卡在了0.99秒处在预期范围，主要时间开销发生在选定节点后的随机模拟。模拟次数通常在5万-15万间。通过天梯对战和自身模拟次数比较从而进行程序的调整优化。

## 经典战局

无

# 总结

总结这次代码的完成，写出代码所用的时间仍然是远少于寻找bug，调试改bug的时间，但通过对bug的处理，队员们对程序的认识，对算法的理解自然也加深了。不断的调试，不断的修正小bug。当然，逻辑思考也是必不可少，通过机器与人的思考相辅相成，最终解决了困扰多日的蛇会断掉的bug。Bug处理掉后，蛇即有了不俗的战力，经过调参测试，最终选定了适合程序的参数值，形成了最终版贪吃蛇bot。

# 参考文献

UCT算法\_大菜包-CSDN博客\_uct算法

<https://blog.csdn.net/yw2978777543/article/details/70799799>

C++程设实验项目三：黑白棋与基于UCT算法的AI\_weixin\_34014277的博客-CSDN博客

https://blog.csdn.net/weixin\_34014277/article/details/94523207?utm\_source=app&app\_version=4.7.0&code=app\_1562916241&uLinkId=usr1mkqgl919blen