

实验1.1

一、背景

你出于兴趣成为了一名南极科考队员，在这一次的科学考察过程中不幸遭遇暴风雪与其他队员走散了，但好消息是暴风雪逐渐减弱了，你重新获得了你的地图定位，并且在走散前你刚刚进行过一次补给，身上装有足够吃 T 天的食物。你需要克服困难，做好规划，在食物消耗完前回到科考站。

二、问题描述

我们将地图简单化为 $M \times N$ 的方格地图，你每天可以移动到与你相邻的上、下、左、右四块方格中的可通行的某一块，请注意，地图（南极大陆）外都是海洋，无法通行。

你在移动过程中应当时刻注意自己的食物储备，你在地图上每移动一格将消耗一天，也会吃掉一天份额的食物，当你身上的食物储备消耗完后，如果你还没有到达科考站或者原地进行补给，你将永远地留在这篇冰封之地。别担心，科考队提前布置了数处补给点，它们已经被标注在了地图上，补给点有充足的物资供应，你可以将身上的食物储备补充满（足够吃 T 天）。简单起见，假定食物补充不需要消耗时间。

地图上有五种格子：

- 0：表示可通行的空地
- 1：表示不可通行的山脉、沟壑等
- 2：表示补给点
- 3：表示你的出发位置
- 4：表示科考站的位置

你能否安全回到科考站？如果能，最短需要多少天？

该任务中，你的目标是：

- 设计一个合适的启发式函数，并证明它是 admissible 的，并论证其是否满足consistent 性质
- 基于上述启发式函数，开发对应的A*算法找到以一个最优解，使得你可以以最短的时间返回科考站
- 设置启发式函数为0，此时A*退化为一致代价搜索算法，比较并分析A*方法带来的优化效果

输入：

详见 `./Astar/input` 共有10个测试文件，形式如下：

第一行三个整数 M ， N ， T ，表示地图的大小 $M \times N$ 以及身上最多可携带的食物储备天数 T
下面 M 行，每行 N 个数字来描述地图

举例：

```
5 4 2
1 0 4 0
1 0 2 1
0 2 0 2
1 0 2 1
0 3 1 0
```

输出：

结果输出到 `./Astar/output` 文件夹下，文件命名为 `output_(x).txt`，其中 `x` 表示对应第 `x` 个输入样例的输出，如对于 `input_0.txt`，输出文件名为 `output_0.txt`。输出形式如下：

一行表示回到科考站的最短天数，若无法回到科考站则输出 `-1`，同时无需输出下一行
一行表示动作的字符串，一共有四种动作，`U`表示向上，`L`表示向左，`R`表示向右，`D`表示向下。

举例：

```
5
URUUU
```

三、作业要求

1. 编程语言限制为 C/C++，写清核心代码注释。代码主体框架已在 `src/Astar.cpp` 中给出，说明：
 1. 补全完成启发式函数定义部分、A*搜索部分等标注有 `TODO` 标记代码。
 2. 给出的代码**任意处均可修改**以方便同学们不受框架约束自行编程，如修改函数返回值类型、修改结构体类型、删除不必要的循环等，只需保证最后程序产生正确的输出文件即可。
2. 报告中需要描述：
 1. 描述你的启发式函数，证明它是 `admissible` 的，并论证其是否满足 `consistent` 性质。
 2. 算法的主要思路；
 3. 与一致代价搜索进行比较，并分析使用 A* 方法带来的优化效果。
3. 严禁抄袭。

实验1.2

一、背景

本次实验要求编写一个中国象棋对弈程序，使用"alpha-beta剪枝"算法，实现人机对弈，使电脑选择走比较好的步骤。

对于博弈问题，考虑极小极大搜索算法。规定：MAX代表程序方，MIN代表对手方，P代表一个棋局。有利于MAX的势态， $f(P)$ 取正值；有利于MIN的势态， $f(P)$ 取负值；势态均衡， $f(P)$ 取零。基于这个前提，博弈双方要考虑：如何产生一个最好的动作，使其可以尽快获胜。

二、问题描述

能否对棋局有一个好的评估是这个算法很关键的一环。我们需要对棋局做出合适的评估，以确定最好的动作。评估的方面有三个，一个是下一步的棋力，第二个是下一步能做什么，第三个是棋子的价值。

下面给出一个棋盘的定义，实验用例依据该定义来设计给定：

```
[
  ['r', 'n', 'b', 'a', 'k', 'a', 'b', 'n', 'r'],
  ['.', '.', '.', '.', '.', '.', '.', '.', '.'],
  ['.', 'c', '.', '.', '.', '.', '.', 'c', '.'],
  ['p', '.', 'p', '.', 'p', '.', 'p', '.', 'p'],
  ['.', '.', '.', '.', '.', '.', '.', '.', '.'],
  ['.', '.', '.', '.', '.', '.', '.', '.', '.'],
  ['P', '.', 'P', '.', 'P', '.', 'P', '.', 'P'],
  ['.', 'C', '.', '.', '.', '.', '.', 'C', '.'],
  ['.', '.', '.', '.', '.', '.', '.', '.', '.'],
  ['R', 'N', 'B', 'A', 'K', 'A', 'B', 'N', 'R']
]
```

r 和 R 分别表示黑方和红方的车（Rook）。
n 和 N 分别表示黑方和红方的马（knight）。
b 和 B 分别表示黑方和红方的象（Bishop）。
a 和 A 分别表示黑方和红方的士（Advisor）。
k 和 K 分别表示黑方和红方的将（King）。
c 和 C 分别表示黑方和红方的炮（Cannon）。
p 和 P 分别表示黑方和红方的卒（Pawn）。
. 表示空位置。

1. 棋力评估（已在代码中给出）

棋力评估是基于棋子所在位置的分析。对每个棋子的位置进行评分，在这里我们给出现成的棋力评估矩阵，每一个值都代表棋盘上的一个位置，里面的值表示对应棋子在该位置的评分，当然，作为参考，也欢迎自己去尝试修改棋力评估矩阵，以获得更好的效果。

```
[ # 将
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, -8, -9, 0, 0, 0, 0, 0, 0, 0,
  5, -8, -9, 0, 0, 0, 0, 0, 0, 0,
  1, -8, -9, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0
],
```

2. 行棋可能性评估（已在代码中给出）

此部分评估是根据棋子下一步的可能动作来判断行棋的优劣。在中国象棋中，理想的行棋是能够捕获价值更高的敌方棋子。特别地，如果下一步动作能够将军，则行棋的评估值将大幅提升。

例(仅供参考):

下一步能将军, 估值+8888

下一步能吃车, 估值+500

下一步能吃马或炮, 估值+100

.....

3. 棋子价值评估 (已在代码中给出)

棋子价值评估考虑的是各棋子的固定价值, 这个价值在棋手间普遍认可。例如, 车的价值通常认为高于卒。

例(仅供参考):

将军: 70

士: 10

象: 30

马: 300

车: 500

炮: 300

卒: 90

每次估值都需要算出程序方棋局的总体价值和对手方棋局的整体价值。用程序方估值-对手方估值作为这个状态下的估值。如果这个估值大于0, 说明程序方占优势; 反之, 说明对手方占优势。

完成好估值后, 开始alpha-beta的剪枝算法。这部分算法老师PPT中已经比较详细了, 这里不再展示。

输入:

详见 `./Alpha_Beta/input` 共有20个测试文件, 包含数据文件极其对应的棋盘图片, 对应10个残局情况, 假定你均为**红棋方**。



输出：

结果输出到 `./Alpha_Beta/output` 文件夹下，文件命名为 `output_(x).txt`，其中 `x` 表示对应第 `x` 个输入样例的输出，如对于 `1.txt`，输出文件名为 `output_1.txt`。输出形式如下：

一个字符（表示你将要移动的棋子） 一个坐标（表示待移动棋子在移动前的位置） 一个坐标（表示待移动棋子在移动后的位置）

举例：

K (4,0) (5,0)

代码框架

代码主体框架已在 `src/` 中给出，其中 `src/node.h` 文件是主题框架，节点类、棋盘类均已给出（有部分接口需要完善），下面是需要完成的部分：

- 合法动作：代码中已经给出车和马的合法动作，可参考这部分代码完成其他部分
- 棋盘分数评估：根据当前棋盘棋子价值和棋子棋力评估棋盘分数

- Alpha-beta剪枝过程

代码框架仅仅是一种简单的实现方式，仅供同学们参考；比如棋盘分数可以根据历史的行动序列进行评估

同学们如果有更好的框架也可以自行实现。

三、作业要求

算法实现

- 阅读并理解极小极大搜索和alpha-beta剪枝算法的相关理论。
- 编写C++代码，实现alpha-beta剪枝算法的棋局搜索。
 - 考虑到完成从头开始的象棋博弈难度较大，助教会给出多张中国象棋残局的输入，给出在当前残局下你下一步应当如何行棋才能获得最大的收益。
 - 象棋博弈并不需要一定完成对局，更关注的应该是棋盘每一步动作是否符合你的棋盘评估方法。
 - 非常欢迎有兴趣的同学完成从头开始的象棋。

实验报告撰写

- 撰写实验报告，详细描述算法实现过程和实验结果。
- 分析alpha-beta剪枝对搜索效率的影响，以及评估函数的设计思路和效果。

实验提交

1. 提交方式：[bb系统](#)中提交
2. 截止日期：5月6日晚23:59
3. 提交的目录树结构如下所示：

```
PB21000001_张三_lab1/
|---Astar
|   |---src
|       |---(your code)
|   |---input
|       |---input_0.txt
|       |---...
|   |---output
|       |---output_0.txt
|       |---...
|---Alpha_Beta
|   |---src
|       |---(your code)
|   |---input
|       |---1.txt
|       |---1.jpg
|       |---...
|   |---output
|       |---output_1.txt
|       |---...
|---report.pdf
```

最后将文件压缩成.zip格式进行提交，**注意**，请大家务必按照目录树结构组织文件，否则可能导致检查脚本读取结果失败。

4. 请务必按时提交实验。