

Quoridor

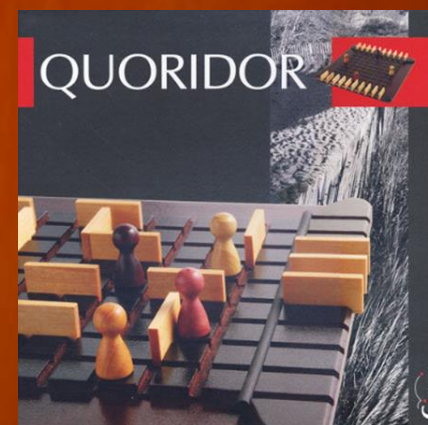
步步为营

P
r
o
j
e
c
t
3

风雨半春长 龙抬头日日丽光 邻塘睡
莲预惊萍 涟漪 踏春不见卧盼离

计算机科学与技术系
191220008

陈南
瞳制



需求分析

壹

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香

算法思路


贰

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香

函数划分

叁

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香



预期效果

肆

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香



用户手册

伍

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香



陆

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼离
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香

壹

需求分析

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

Quoridor（步步为营）

要求：

在Quoridor现有的的游戏规则框架下，实现一个具有一定行棋策略的虚拟玩家，并要求尽可能地击败其他虚拟玩家，获取胜利！



操作规则：

(1) 一次仅允许移动一步或放置一块木板，否则视为违规操作。

(2) 面前是对方小人时，若其后没有木板，则移动到对方背后，否则视为违规操作。

(3) 若10块木板用尽，则只允许进行移动操作，否则视为违规操作。

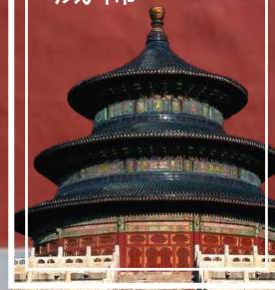
(4) 木板长度为2，且仅能竖直或水平放置，否则视为违规操作。

(5) 放置的木板不允许将对方的所有获胜路径堵死。



▲
需求分析
▼

雪入春分省见稀
半开桃李不胜威



操作规则：

(6) 当小人位于对方的唯一路径上时，只允许移动，不允许放置木板，否则视为违规操作。

(7) 每次必须进行操作，不能无操作，否则视为违规操作。

(8) 所有操作不允许超出棋盘边界，否则视为违规操作。

(9) 违规操作3次则直接判负。

(10) 单次决策超过1s视为超时，直接判负。



▲
需求分析
▼

雪入春分省见稀
半开桃李不胜威



实现要求：

1、能够完整地记录当前棋盘的状态，包括：

- (1) 己方的位置、敌方的位置；
- (2) 所有木板的位置；
- (3) 己方所剩木板数目、敌方所剩木板数目；
- (4) 己方胜利的路径、敌方胜利的路径（从短到长）；
- (5) 己方违规次数、敌方违规次数（作用待定）。

2、具有一定的算法：

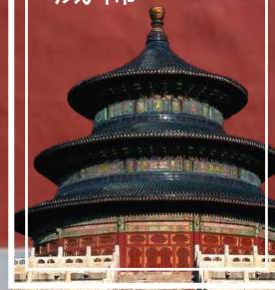
- (1) 能够根据当前棋盘的局势，决定移动还是放置木板。
- (2) 能够计算出尽可能有利的移动方向或者木板放置的位置。
- (3) 能够计算出己方和敌方胜利的路径和步数（从短到长）。
- (4) 能够判断放置的木板是否会堵死敌方。

3、满足一些限制：

- (1) 单步计算时间不得超过1s。
- (2) 能够判断操作是否合法，且不能违规3次。
- (3) 尽可能在100步内取得胜利，否则按计算时间判胜负。
- (4) 放置的木板不能堵死敌方（和己方？）。

需求分析

雪入春分省见稀
半开桃李不胜威





贰

算法思路

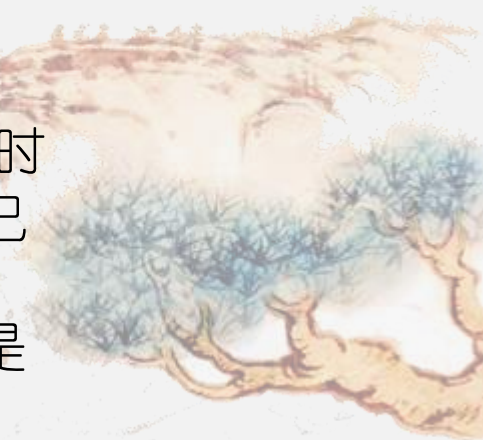
春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥



游戏类型：

要想确定算法，必须先确定游戏类型，再从适合该游戏的算法中进行选择。

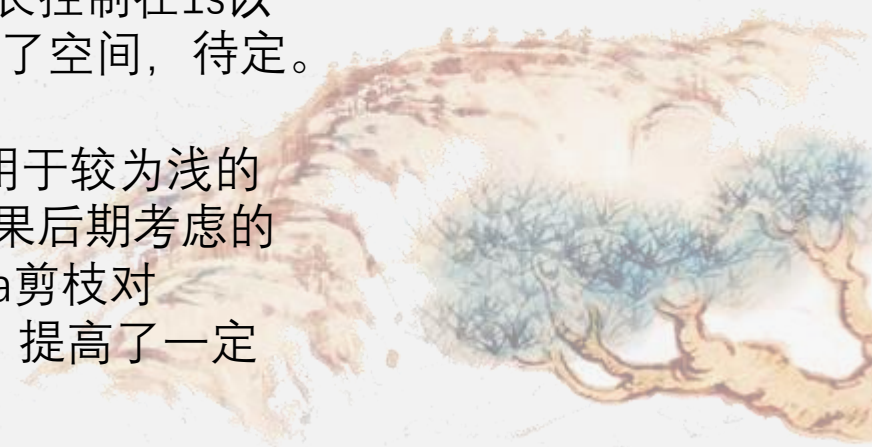
通过了解Quoridor的规则，不难发现，该游戏是一个典型的Combinatorial Game，自然满足：zero-sum、perfect/fully information、deterministic、sequential、discrete，也就是说这个场景必须是能分出输赢（不能同时赢）、游戏的信息是完全公开的（不像打牌可以隐藏自己的手牌）、确定性的（每一个操作结果没有随机因素）、顺序的（操作都是按顺序执行的）、离散的（没有操作是一种连续值）。



搜索算法考虑：

对于Combinatorial Game，部分可行的搜索算法列举如下：

- 1、DFS/BFS：首先排除DFS，因为DFS的搜索方式过于盲目，即便进行了剪枝，也不一定能将搜索时长控制在1s以内。相较于DFS，BFS则节约了时间，但浪费了空间，待定。
- 2、Alpha-Beta/minimax+剪枝：minimax适用于较为浅的树的搜索，对于较深的树则效率非常低，如果后期考虑的层数不多，则可能考虑minimax。Alpha-Beta剪枝对minimax进行了改进，除去了不必要的搜索，提高了一定效率，不过仍有效率不高的可能，待定。



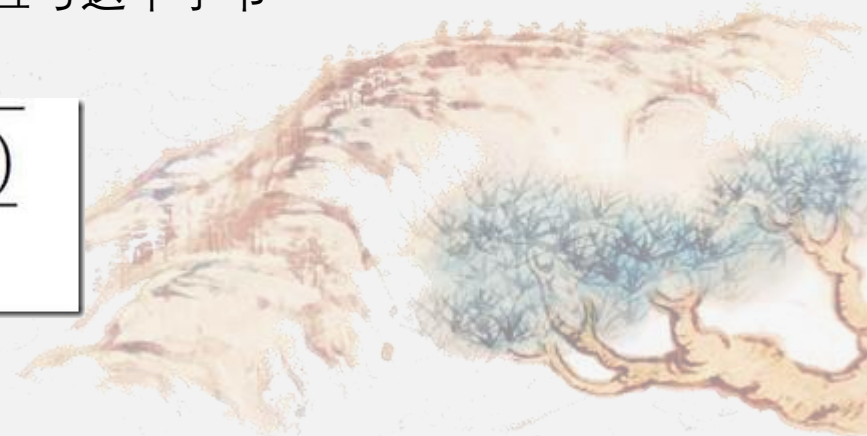
搜索算法考虑：

3、A*：其实是对BFS的一种改进，待定。

4：MCTS：可以采用UCB算法，在选择子节点的时候优先考虑没有探索过的，如果都探索过就根据得分来选择，得分不仅是由这个子节点最终赢的概率来，而且与这个子节点玩的次数成负相关。

$$\arg \max_{v' \in \text{children of } v} \frac{Q(v')}{N(v')} + c \sqrt{\frac{2 \ln N(v)}{N(v')}}}$$

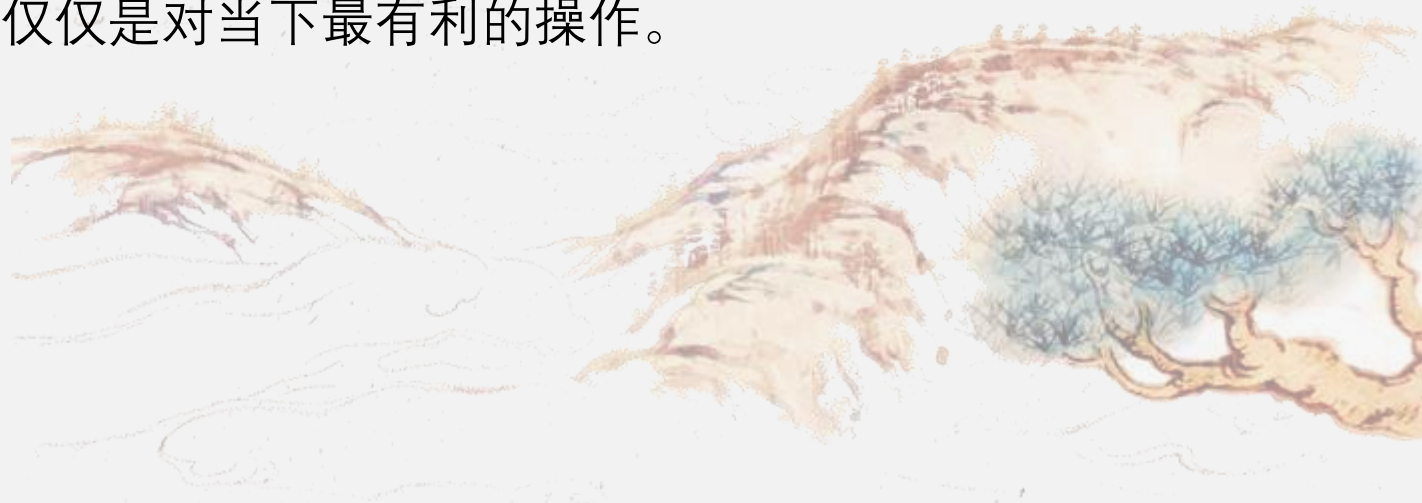
不过MCTS仍很可能会超时，待定。





针对“贪心算法”：

为了较好的应对使用“贪心算法”策略的玩家，需要进行适当的预判，并采用适合的算法来应对，如：Alpha-Beta算法或minimax算法，可以有效地利用对方贪心算法的特点，实现预判后，进行对将来最有利的操作，而非仅仅是对当下最有利的操作。



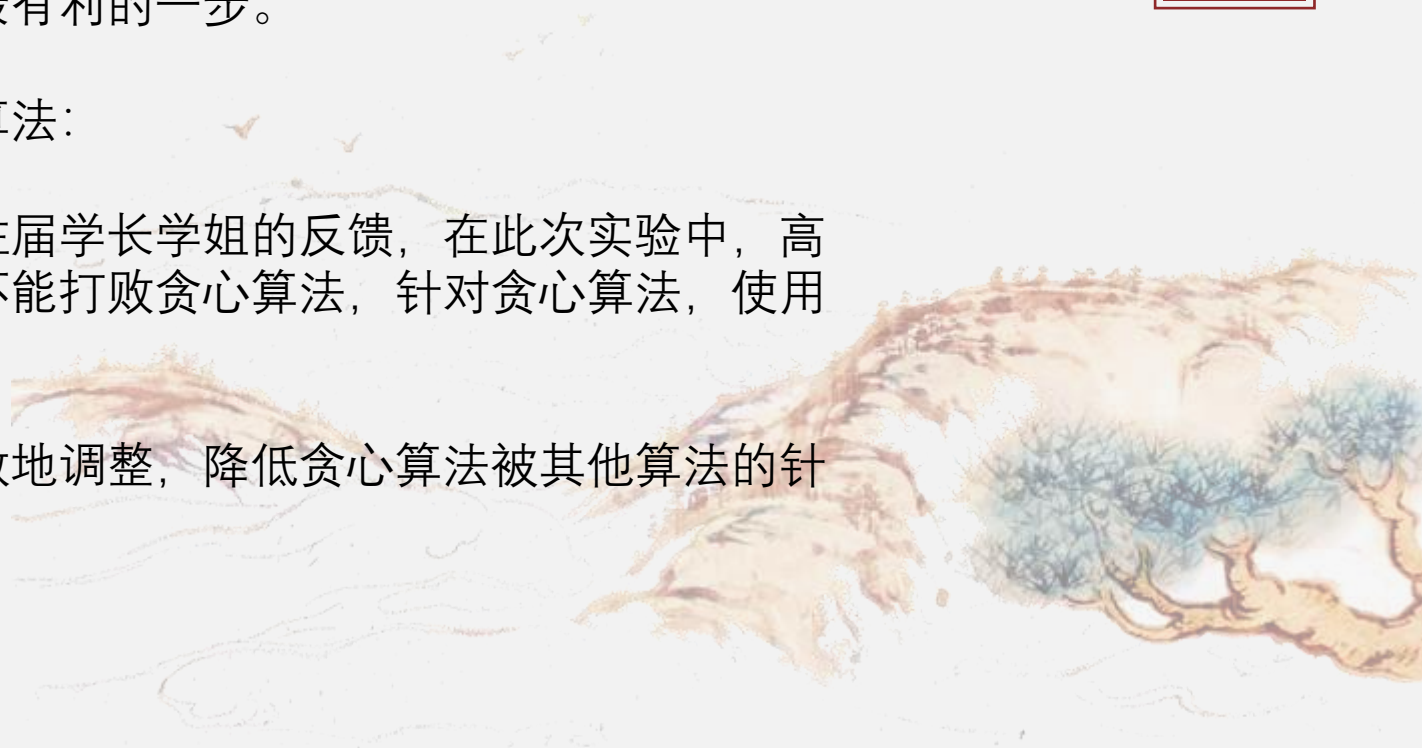
当前选择：

当前暂时选择BFS进行搜索，并采用贪心算法，即每一步都是对当前局势最有利的一步。

为何不选择其他算法：

根据其他同学和往届学长学姐的反馈，在此次实验中，高级的算法常常并不能打败贪心算法，针对贪心算法，使用BFS已经足够。

可以通过评估函数地调整，降低贪心算法被其他算法的针对。





我的BFS:

使用队列的方式。用Node queue[100]结构数组储存节点，再用Node path[10000]储存最短路径。

将所有搜索路径时所处的位置存入队列，然后从队列中弹出一个节点，每判断完弹出的节点的所有相邻节点后，将符合条件的相邻节点存入队列，然后继续从队列弹出节点，以此类推，直到有节点达到终点。



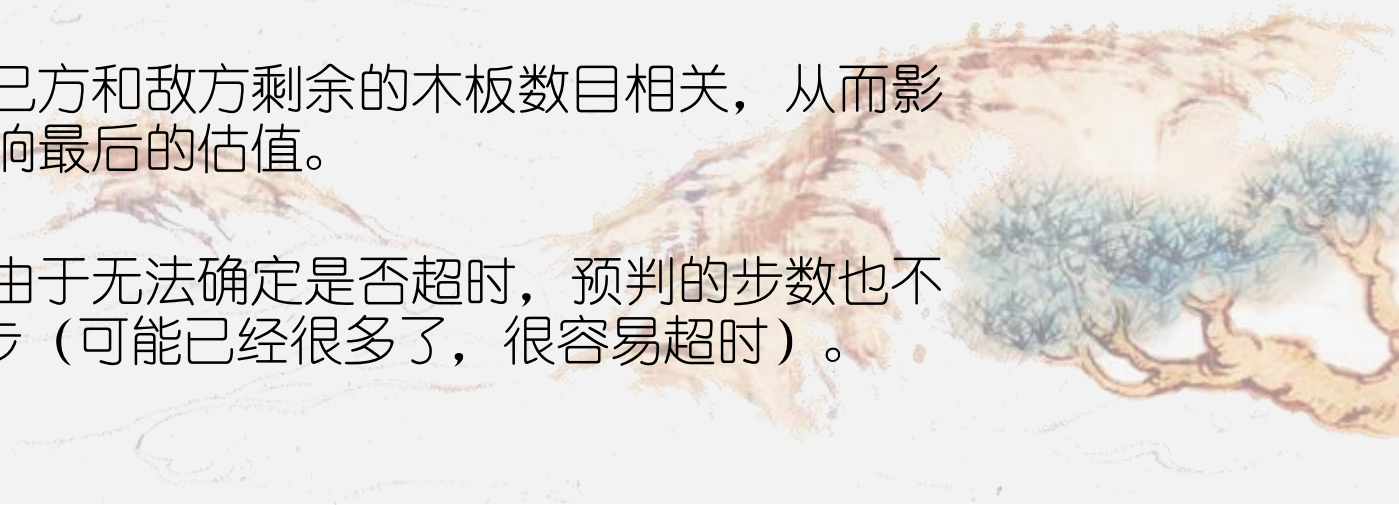


评估函数：

当前的估值函数是以己方胜利最短路径步数和敌方最短胜利路径步数等为自变量的函数（后期可能拓展：路径不一定是最佳路径，可能是多条路径估值的运算，但权重不同，难点就在权重的分配和公式的形式）。

该估值函数还与己方和敌方剩余的木板数目相关，从而影响权重，进而影响最后的估值。

后期可能拓展：由于无法确定是否超时，预判的步数也不确定，暂定2~3步（可能已经很多了，很容易超时）。



评估函数：

判断放木板还是移动：

我设立了两个评估函数（score_1和score_2），一个适用于对移动的评估，一个适用于对木板放置的评估。以及针对特定情况的加分处理。

$$\text{score_1} = 22.0 * (\text{double})\text{enemy_path_min_step_number} + 1.0 * (\text{double})\text{enemy_block_left} - 5.0 * (\text{double})\text{my_path_min_step_number} - 3.0 * (\text{double})\text{my_block_left};$$

$$\text{score_2} = 2.0 * (\text{double})\text{enemy_block_min_step_number} - 1.0 * (\text{double})\text{enemy_block_left} - 5.0 * (\text{double})\text{my_block_min_step_number} + 4.0 * ((\text{double})\text{my_block_left} - 1) + 40.0 * ((\text{double})\text{enemy_block_min_step_number} - (\text{double})\text{enemy_path_min_step_number});$$

（该公式中的参数和公式格式直接决定了策略的好坏，进而决定对战的输赢）



评估函数：

```

if (my_block_min_step_number - my_path_min_step_number > enemy_block_min_step_number -
enemy_path_min_step_number)
    score_1 = score_1 + 50;
if (my_path_min_step_number < 4 && my_path_min_step_number < enemy_path_min_step_number)
    score_1 = score_1 + 10000;
if (enemy_path_min_step_number < 4 && my_path_min_step_number >= enemy_path_min_step_number)
    score_2 = score_2 + 10000;
if (my_path_min_step_number > enemy_path_min_step_number + 1)
    score_2 = score_2 + 10;
else if (my_path_min_step_number > enemy_path_min_step_number)
    score_2 = score_2 + 15;
if (my_path_min_step_number > enemy_path_min_step_number + 3 && my_block_min_step_number -
my_path_min_step_number <= enemy_block_min_step_number - enemy_path_min_step_number)
    score_2 = score_2 + 25;
if (my_block_min_step_number - my_path_min_step_number + 1 < enemy_block_min_step_number -
enemy_path_min_step_number)
    score_2 = score_2 + 35;
if (my_block_min_step_number - my_path_min_step_number <= enemy_block_min_step_number -
enemy_path_min_step_number && my_path_min_step_number > enemy_path_min_step_number + 2)
    score_2 = score_2 + 50;

```



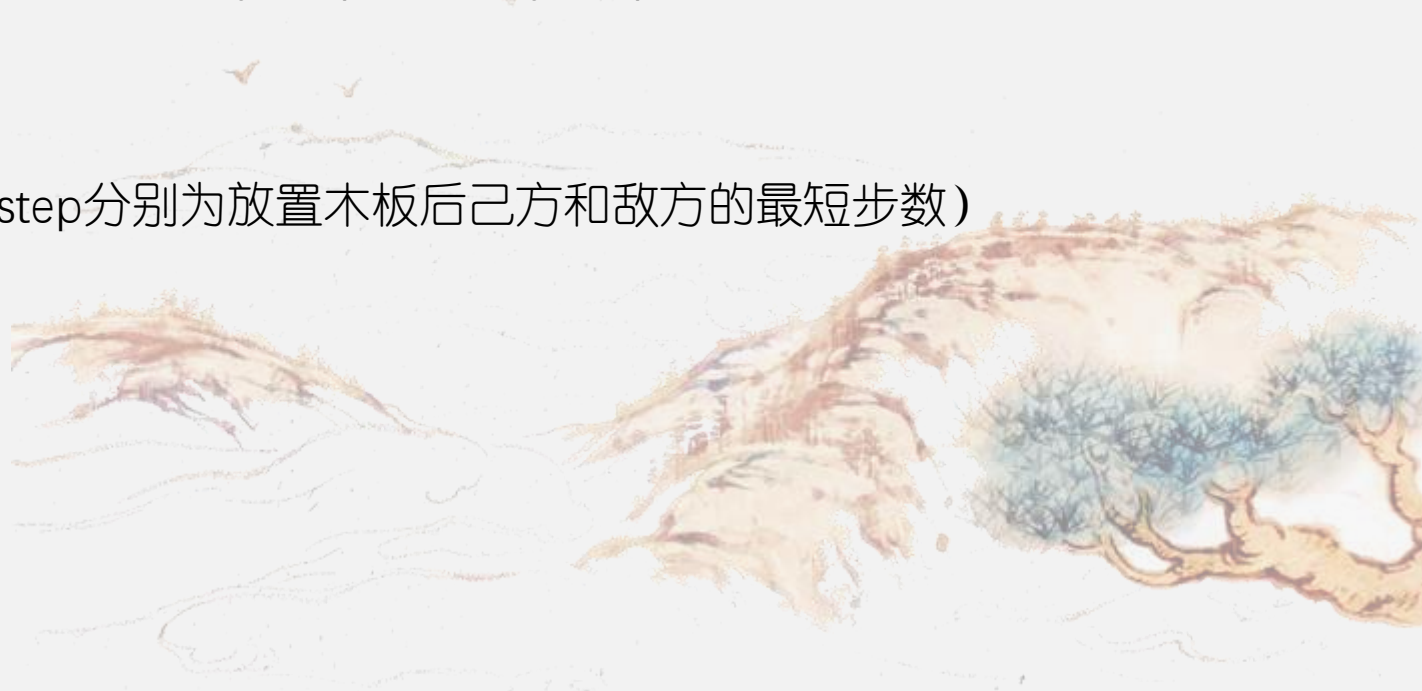


评估函数：

判断木板放置的位置： $a * \text{enemy_step} - b * \text{my_step}$

(取 $a=1$, $b=1$)

(my_step 和 enemy_step 分别为放置木板后己方和敌方的最短步数)



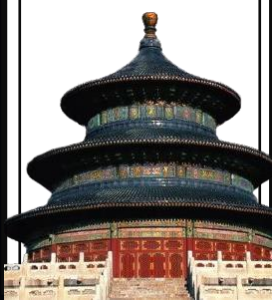
叁

函数划分

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

函数划分

雪入春分省见稀
半开桃李不胜威

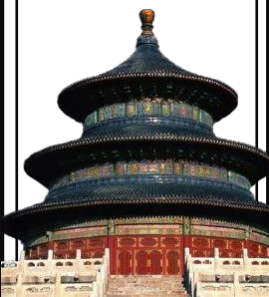


变量说明:

```
int targetY = 0; // 目标方向
QuoridorUtils::Location min_Loc; // 最佳移动的坐标
QuoridorUtils::BlockBar min_block_set; // 最佳的放置木板的坐标
QuoridorUtils::Location next_Loc; // 下一步移动的坐标
QuoridorUtils::BlockBar next_block_set; // 下一步放置木板的坐标
int path_status = -1; // 是否移动
int block_status = -1; // 是否放置木板
int my_path_min_step_number = 0; // 己方最短的获胜步数
int enemy_path_min_step_number = 0; // 敌方最短的获胜步数
int my_block_min_step_number = 0; // 放置木板后己方最短的获胜步数
int enemy_block_min_step_number = 0; // 放置木板后敌方最短的获胜步数
std::vector<QuoridorUtils::BlockBar> blocks; // 存储所有的木板
int head = 0, tail = 0; // 节点队列的首尾下标
Node queue[100], path[10000]; // 节点和路径的结构
int board[9][9] = { 0 }; // 棋盘数组，用于标记寻路时岛国的位置
int my_block_left = 10; // 己方剩余木板数
int enemy_block_left = 10; // 敌方剩余木板数
int illegal = 0; // 是否犯规
int my_illegal_times = 0; // 犯规次数
```

函数划分

雪入春分省见稀
半开桃李不胜威



函数说明:

```
void my_strategy(const QuoridorUtils::Location& myLoc, const  
QuoridorUtils::Location& enemyLoc);  
//总走棋策略函数
```

```
void direction_check(bool directions[], const QuoridorUtils::Location&  
myLoc, const QuoridorUtils::Location& enemyLoc);  
// 判断移动是否合法
```

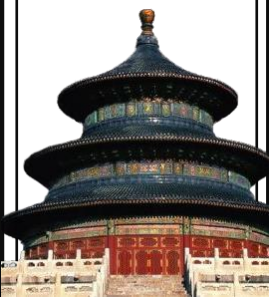
```
int path_seek(const QuoridorUtils::Location& myLoc, const  
QuoridorUtils::Location& enemyLoc);  
// 寻找最短路径
```

```
bool block_check(const QuoridorUtils::BlockBar& block_set, const  
QuoridorUtils::Location& myLoc, const QuoridorUtils::Location&  
enemyLoc);  
// 判断放置的木板是否合法
```

```
int block_seek(const QuoridorUtils::Location& myLoc, const  
QuoridorUtils::Location& enemyLoc);  
// 寻找最佳的放置的木板
```


函数划分

雪入春分省见稀
半开桃李不胜威



函数说明:

```
void board_initialize();  
// 涂图棋盘初始化
```

```
Node* pop();  
// 弹出队列
```

```
void push(QuoridorUtils::Location Loc, Node* next);  
// 加入队列
```

```
double evaluation(double my_path_number, double my_block_number,  
double enemy_path_number, double enemy_block_number);  
// 评估函数（已废弃，现使用另一个）
```

```
Step nextStep(const ChessboardChange& newChange) ;  
// 下一个操作
```

```
void restart() ;  
//清空当前数据，为下一轮对战做准备
```

预期效果

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

肆

风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分

风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分

风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分



虽然发现运气占比较大，但仍可通过策略将获胜概率一定幅度地提高。
所以预期效果即为尽可能战胜更多的AI。

预期效果

雪入春分省见稀
半开桃李不胜威



```

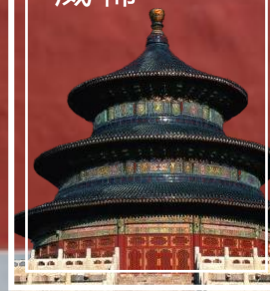
game status: Ok; my loc: (2, 9); enemy loc: (5, 2) -> move to: (1, 9)    0.036
game status: Ok; my loc: (1, 9); enemy loc: (5, 1) -> move to: (1, 8)    0.017
game status: Ok; my loc: (1, 8); enemy loc: (4, 1) -> move to: (2, 8)    0.017
game status: Ok; my loc: (2, 8); enemy loc: (3, 1) -> move to: (3, 8)    0.017
game status: Ok; my loc: (3, 8); enemy loc: (3, 2) -> move to: (3, 7)    0.017
game status: Ok; my loc: (3, 7); enemy loc: (2, 2) -> move to: (3, 6)    0.017
game status: Ok; my loc: (3, 6); enemy loc: (1, 2) -> move to: (2, 6)    0.017
game status: Ok; my loc: (2, 6); enemy loc: (1, 3) -> move to: (1, 6)    0.018
game status: Ok; my loc: (1, 6); enemy loc: (2, 3) -> move to: (1, 5)    0.041
game status: Ok; my loc: (1, 5); enemy loc: (3, 3) -> move to: (1, 4)    0.015
game status: Ok; my loc: (1, 4); enemy loc: (3, 4) -> move to: (2, 4)    0.012
game status: Ok; my loc: (2, 4); enemy loc: (1, 4) -> move to: (3, 4)    0.01
game status: Ok; my loc: (3, 4); enemy loc: (1, 5) -> move to: (3, 3)    0.008
game status: Ok; my loc: (3, 3); enemy loc: (1, 6) -> add BlockBar: [(1, 5); (1, 7)]    0.008
game status: Ok; my loc: (3, 3); enemy loc: (1, 5) -> move to: (2, 3)    0.008
game status: Ok; my loc: (2, 3); enemy loc: (2, 5) -> move to: (1, 3)    0.008
game status: Ok; my loc: (1, 3); enemy loc: (2, 6) -> add BlockBar: [(2, 5); (2, 7)]    0.008
game status: Ok; my loc: (1, 3); enemy loc: (2, 5) -> move to: (1, 2)    0.007
game status: Ok; my loc: (1, 2); enemy loc: (3, 5) -> move to: (2, 2)    0.007
game status: Ok; my loc: (2, 2); enemy loc: (3, 6) -> move to: (3, 2)    0.006
game status: Ok; my loc: (3, 2); enemy loc: (3, 7) -> move to: (3, 1)    0.006
Game result:Win
Save log file to log_20200514_011227_749.csv
Clean your player.
All done.
result: victory(3), defeated(0).

```

C:\南\程序设计基础实验\第十~十三周\需要文件\Quoridor\Debug\QuoridorClient.exe (进程 12932)已退出, 代码为 0。
按任意键关闭此窗口. . .

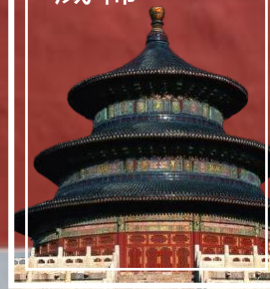
预期效果

雪入春分省见稀
半开桃李不胜威



预期效果

雪入春分省见稀
半开桃李不胜威



```
C:\南\程序设计基础实验\第十~十三周\需要文件\Proj3_server_1.051>java -jar server.jar -p 19330 -n 2 -r 3
Server Version: 1.1
Args: [-p, 19330, -n, 2, -r, 3]
```

```
-----
Waiter is waiting for the players to sit down.
Waiter's work is done.
Contest start at: 2020-05-14 01:12:24
```

```
-----***-----
*****
Knockout Round 1 :
*****
[3 : 0] 191220008 defeat-->
```

```
-----
CHAMPION: QuoridorPlayer{name='191220008'}
```

```
-----***-----
Contest end at: 2020-05-14 01:12:27
```




伍

用户手册

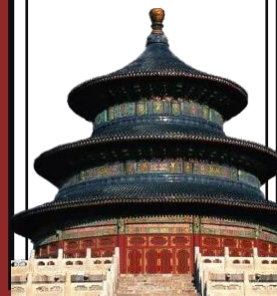
春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥



详见用户手册

用户手册

雪入春分省见稀
半开桃李不胜威



旅行

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

风雷送暖入中春，桃柳着装日日新。
赤道金阳直射面，白天黑夜两均分。

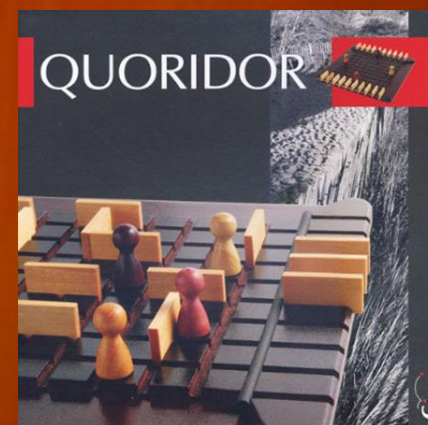
Thanks

P
r
o
j
e
c
t
3

风雨半春长 龙抬头日日丽光 邻塘睡
莲预惊萍 涟漪 踏春不见卧盼离

191220008
计算机科学与技术系

陈南
瞳制





旅行

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

雪入春分省见稀 半开桃李不胜威 应
惭落地梅花识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従
今造物尤难料 更暖须留御腊衣 雪入
春分省见稀 半开桃李不胜威 应惭落
地梅花识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従
今造物尤难料 更暖须留御腊衣 雪入
春分省见稀 半开桃李不胜威 应惭落
地梅花识 却作漫天柳絮飞



旅行

风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分

旅

菜花青径香

翠正好，笑扬牵纤

发发更藏，宜三伴窗

春分艳阳天，齿儿戏

不见卧盼离

预惊萍，涟漪踏春

日日丽光，邻塘睡莲

风雨半春长，龙抬头

行

菜花青径香

翠正好，笑扬牵纤

发发更藏，宜三伴窗

春分艳阳天，齿儿戏

不见卧盼离

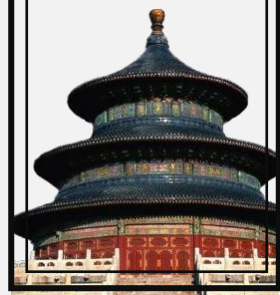
预惊萍，涟漪踏春

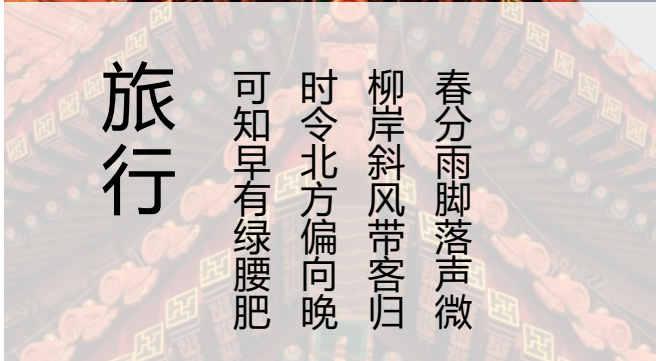
日日丽光，邻塘睡莲

风雨半春长，龙抬头

旅行

雪入春分省见稀
半开桃李不胜威





旅行

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

旅行

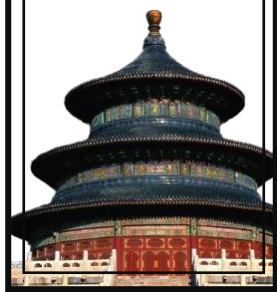
春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

旅行

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

旅行

雪入春分省见稀
半开桃李不胜威



▲ 旅行

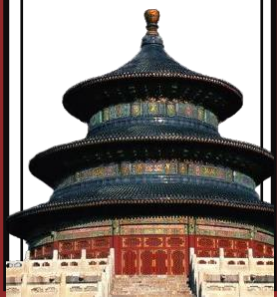
风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼篱
春分艳阳天 齿儿戏
发发更藏 宜三伴窗
翠正好 笑扬 牵纤
菜花青径香



风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分
风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分
风雷送暖入中春
桃柳着装日日新
赤道金阳直射面
白天黑夜两均分

旅行

雪入春分省见稀
半开桃李不胜威



QINGMING

清明

雪入春分省见稀 半开桃李不胜威 应惭
落地梅花识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞



旅行

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

风雷送暖入中春，桃柳着装日日新。
赤道金阳直射面，白天黑夜两均分。

旅行

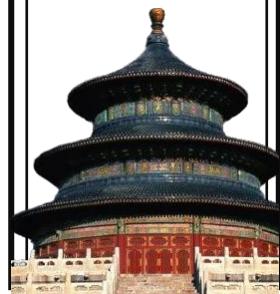
风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼篱
春分艳阳天 齿儿戏
发发更藏 宜三伴窗
翠正好 笑扬 牵纤
菜花青径香

春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥



旅行

雪入春分省见稀
半开桃李不胜威



旅行

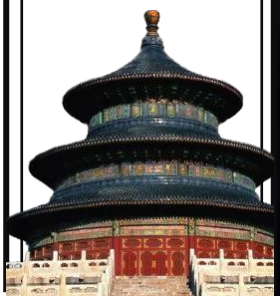
雪入春分省见稀 半开桃李不胜威 应惭
落地梅花识 却作漫天柳絮飞

不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞



旅行

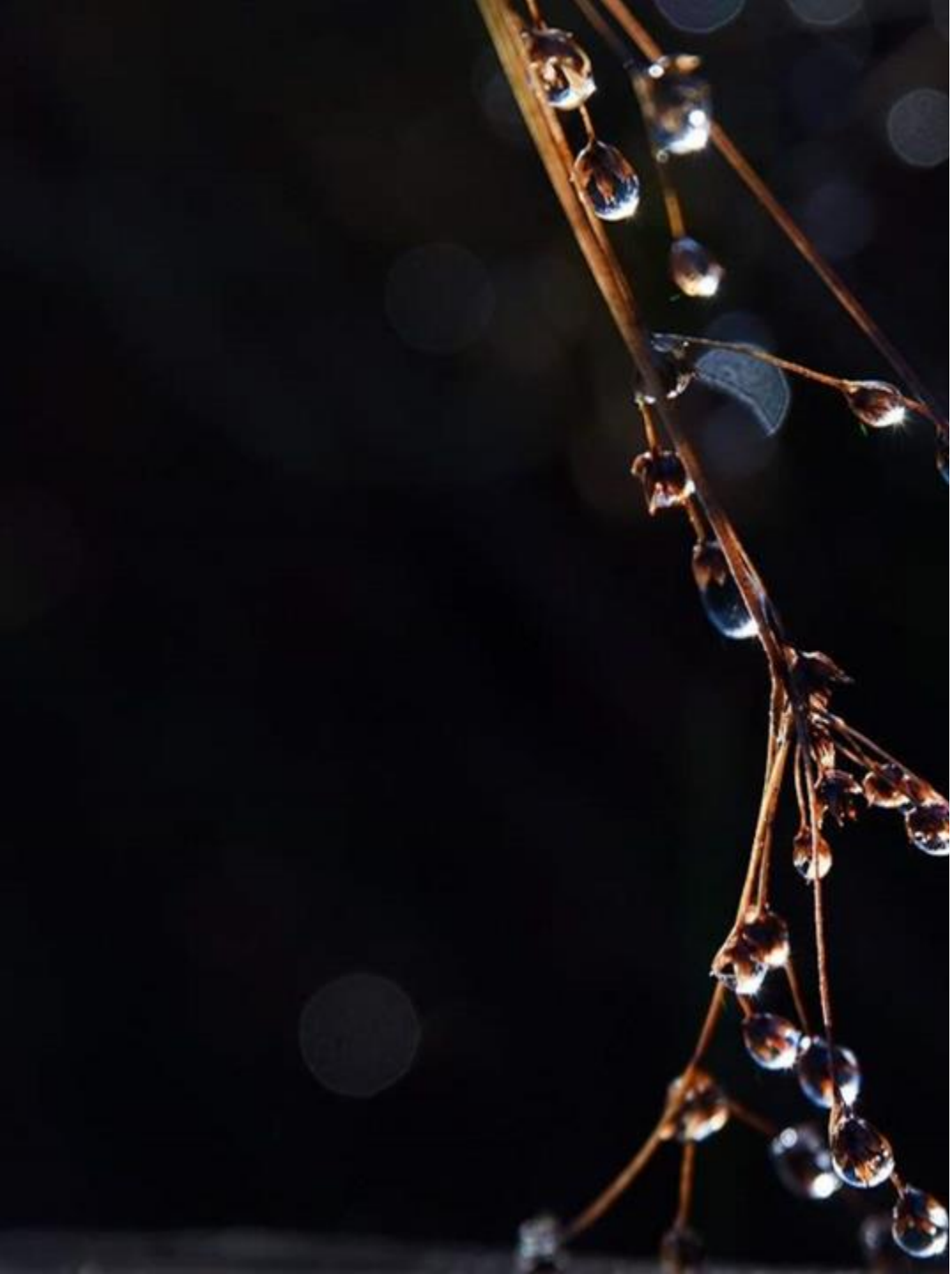
雪入春分省见稀
半开桃李不胜威



旅行

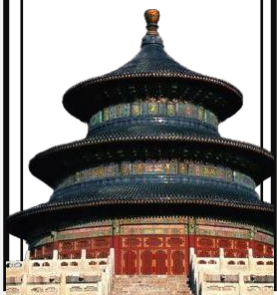
春分雨脚落声微
柳岸斜风带客归
时令北方偏向晚
可知早有绿腰肥

雪入春分省见稀 半开桃李不胜威 应惭
落地梅花识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞
不分东君专节物 故将新巧发阴机 従今
造物尤难料 更暖须留御腊衣 雪入春分
省见稀 半开桃李不胜威 应惭落地梅花
识 却作漫天柳絮飞



旅行

雪入春分省见稀
半开桃李不胜威



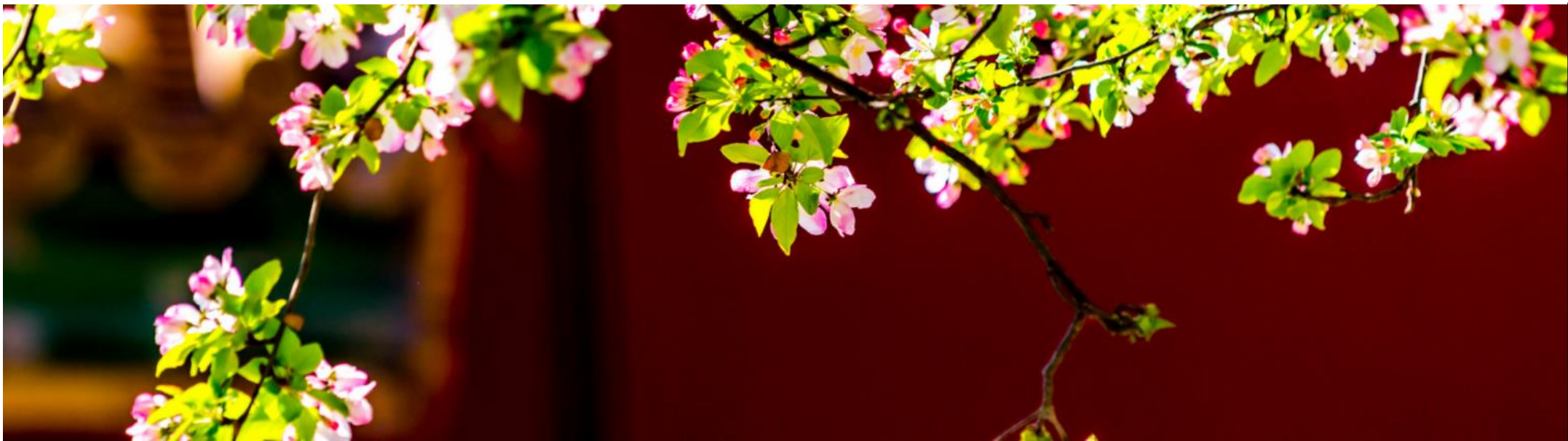
行

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼篱
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香



旅

风雨半春长 龙抬头
日日丽光 邻塘睡莲
预惊萍 涟漪 踏春
不见卧盼篱
春分艳阳天 齿儿戏
发发更藏 宜兰伴窗
翠正好 笑扬 牵纤
菜花青径香



未央

陈南
瞳制

风雨半春长 龙抬头日日丽光 邻塘睡
莲预惊萍 涟漪 踏春不见卧盼离

新中式排版系列