人工智能基础: 编程作业1实验报告(1)

> 姓名：王博
> 学号：PB16020870
> 实验名称：A*搜索问题

1. 实验要求概述

- 在用二维数组表示的，大小设置为 18*25 的迷宫上，指定起点和终点，计算路径并输出动作序列。
- 输出时将花费的时间（以 s 为单位），动作序列，总步数输出到文件。字母大写， 字母之间无空格。
- 使用A* 和 IDA* 两种算法来实现。

2. 实验环境

- 操作系统：Windows subsystem for linux, Ubuntu 18.04
- 编译器：gcc 7.3.0
- 编译工具：CMake 2.8 +

3. 文件目录和编译运行说明

3.1 文件结构

```
.
├── CMakelists.txt  //CMake
├── bin
│   ├── astar           //A*算法执行文件
│   └── idastar         //IDA*算法执行文件
├── doc
│   ├── <小/大地图输出>  //程序输出文output.txt的结果备份
│   └── lab1.md         //实验报告源码
├── include         //include文件目录
```

```
│   ├── Astar.hpp
│   ├── FileIO.hpp
│   ├── IDAstar.hpp
│   ├── Viz.hpp
│   ├── include_all.hpp
│   └── my_debug.hpp
├── inputs          //输入输出文件目录
│   ├── input.txt       //输入地图文件
│   └── input2.txt      //第二个输入地图文件
├── main_astar.cpp      //A*算法主文件
├── main_idastar.cpp    //IDA*算法主文件
└── src
    ├── Astar.cpp       //A*算法代码
    ├── FileIO.cpp      //文件IO代码
    ├── IDAstar.cpp     //IDA*算法代码
    └── Viz.cpp         //屏幕调试输出代码
```

3.1 编译运行办法

- 首先，切换至本项目目录下，然后输入如下命令完成编译：

```
cmake .
make
```

- make成功后，可执行文件在bin/目录下。
  - A * 算法对应程序文件astar， IDA * 算法对应可执行文件idastar.
  - 这两个程序的运行方式都是（i是竖向，j是横向）：

```
执行程序 输入文件路径 起点i 起点j 终点i 终点j
```

- 运行命令举例：

```
./bin/astar inputs/input.txt 1 0 16 24
./bin/astar inputs/input.txt 1 3 16 24
./bin/astar inputs/input.txt 7 2 16 24
./bin/astar inputs/input.txt 14 1 16 24

./bin/idastar inputs/input.txt 1 0 16 24
./bin/idastar inputs/input.txt 1 3 16 24
./bin/idastar inputs/input.txt 7 2 16 24
./bin/idastar inputs/input.txt 14 1 16 24

./bin/astar inputs/input2.txt 1 0 28 59
./bin/idastar inputs/input2.txt 1 0 28 59
```

- 可执行程序会读取指定的输入文件，并将输出写到当前目录下的output_A.txt和output_IDA.txt.
- 可执行程序还会将运行结果的地图展示输出到标准输出（屏幕）。

4. 实验过程

4.1 算法设计和关键代码

4.1.1 启发式函数

常见的选择是欧氏距离和曼哈顿距离。而本问题由于给定走法，启发式函数选择比较简单，因为由于按照行走规则，无障碍或者障碍不造成折返的情况下，曼哈顿距离和真实行走距离想等。这样的话，在没有更多信息和预处理的情况下，曼哈顿距离很可能是一个最佳的选择。所以我选择曼哈顿距离：

$$h(i,j) = Abs(i - i_{target}) + Abs(j - j_{target})$$

4.1.2 A*算法设计

A*算法编写过程中，需要注意的点是：

- 注意A*在比较的是$f(x) = h(x) + g(x)$。
- 需要将$f(x)$的值，和地图上的点配对放入优先队列。
- 放入优先队列之前需要判断是否已经放入，以及是否比放入的更优。原因：
  1. 我的前驱节点信息不在优先队列里，而是在另一个二位数组里。如果不判重，前驱结点就会被随便更新，可能丢失最优解的路径。
  2. 由于我的$g(x)$值也是存储在一个二维数组里的，所以判重和判优都只需要常数时间，成本可忽略。
- 调试时，可以通过二维数组标记A*算法的状态，然后绘制所有地图点的状态来debug。正常情况下，应该看到菱形的等距离边界。
- 默认的优先队列，是大元素在前。所以，可以在存入优先队列时取负值。
- 运行A * 算法之前，需要初始化如下内容：
  - 保存前驱方向(L,R,U,D)的二维数组
  - 保存地图上某点$g(i,j)$的二维数组
  - 保证地图上可行走区域都是0（为了简单起见，在A * 算法运行时，我直接用-1标记地图上的关闭列表了。算法结束时要复原为0.）

4.1.3 IDA * 算法设计

IDA * 算法编写过程中，需要注意的点：

- 外层循环必须从$h_{i,j}$开始，每次增加最小可能单位。对于本实验，每次路径长度增加一定是2的倍数。原因：
  - 如果 $h_{i,j}$ 是奇数，则所有可能的路径长度都是奇数；
  - 如果 $h_{i,j}$ 是偶数，则所有可能的路径长度都是偶数。
- 在DFS的过程中，节点访问顺序的先后，不影响结果的正确性，但是可能对性能有影响。
- IDA * 如果不剪枝，对于自由空地或者出现折回的路径，搜索空间会爆炸。关于算法的优化，在4.3节算法优化中说明。

4.2 算法优化

4.2.1 A * 算法优化

- “关闭列表”写在地图上。这样每次在查找_{i, j}是否是关闭节点时，只需要O(1)时间。这是一个时间优化。
- 减少优先队列中的元素个数。这个已经做了，通过判重和判优。由于前驱方向和g(i,j)写在另一个二维数组上，所以判重判优也是O(1)时间。这是一个时间优化，对于优先队列来说也是空间优化。

4.2.2 IDA * 算法优化

- 通过$g_{best}(i,j)$地图保证IDA*走过的路线是从起点到当前点的最优路线。
  - 做法：在初始时，初始化$g_{best}(i,j)$地图，起点之外的点的$g_{best}(i,j)$都是很大的正数。一次寻路中，迭代加深DFS的过程中，DFS检查到达当前点的$g_{i,j}$是否大于$g_{best}(i,j)$：
    - 如果小于，说明代码写错了，逻辑Bug；
    - 如果等于，说明DFS到达这里是最优路线，值得继续；
    - 如果大于，说明DFS到达这里之前绕弯了，剪掉此枝。
  - $g_{best}(i,j)$在整个寻路期间都不恢复初始值，其信息在多次DFS之间传递。
  - 正确性证明：
  - 这是一个时间优化。
- 在一轮DFS的过程内，通过“行走脚印”标记，避免在自由空间寻路时的路径穷举。
  - 做法：所有探明过的点都是障碍物，本轮DFS不再穿越这些探明过的点。本轮DFS结束后，清除“行走脚印”标记。

◦ 举例：如图黑点所表示的路径是在该 3*8 的自由空间内的一条路径。但是同样是最优的，穿过该区域的路径有很多很多。为了避免DFS做无意义穷举，而只是希望O(3 * 8) 的时间内就标记好该区域内的所有$g_{best}(i,j)$的值，只需要将走过的点视为障碍物即可。

```
  .
# . # # # # # # # #
# . . .         #
#     . . .     #
#         . . . #
# # # # # # # . #
              .
```

```
  #
# # # # # # # # # #
# # # #           #
#     # # # . . . #
#         # # # # #
# # # # # # # # # #
              #
```

```
  #
# # # # # # # # # #
# # # #           #
#     # # # # # # #
#       . # # # # #
# # # # # # # # # #
              #
```

……

在 O(3 * 8) 的时间内，所有满足本轮DFS对$f_{i,j}$限制的点，都会被遍历到并标记上$g_{best}(i,j)$值。

◦ 正确性证明：
◦ 这是一个时间的优化。

4.3 算法复杂度分析

- 优化过的 A * 算法：
  - ◦ 最好情况：路径较窄，岔路较少，没有大量障碍物造成的折返。优先队列里的元素个数几乎不变。
    - 最好时间复杂度：$O(f_{i,j})$
    - 最好空间复杂度(优先队列)：$O(1)$
    - 总空间复杂度：$O(\text{地图大小})$
  - ◦ 最坏情况：路径较宽，大量出口狭小但是却面积大的场地，很多造成折返的障碍物，岔路。优先队列里的元素个数线性增长。如果使用二叉堆实现优先队列：
    - 最坏时间复杂度：$O\{f_{i,j}log(O(f_{i,j}))\}$
    - 最坏空间复杂度(优先队列)：$O(f_{i,j})$
    - 总空间复杂度：$O(\text{地图大小})$
- 原始的 IDA* 算法：
  - ◦ 最好情况：岔路较少，没有大量障碍物造成的折返。前一两轮就到达终点了。
    - 最好时间复杂度：$O(f_{i,j})$
    - 最好空间复杂度：$O(f_{i,j})$
  - ◦ 最坏情况：最坏情况：路径较宽，大量出口狭小但是却面积大的场地，很多造成折返的障碍物，岔路。未经过剪枝的 IDA * 的搜索空间呈阶乘上升。
    - 最坏时间复杂度：$O(f_{i,j}!)$，过大。
    - 最坏空间复杂度：$O(f_{i,j})$

- 剪枝优化后的 IDA* 算法：
  - 最好情况：同原始的 IDA* 算法。
    - 最好时间复杂度：$O(f_{i,j})$
    - 最好空间复杂度：$O(f_{i,j} + 地图大小)$
  - 最坏情况：路径较宽，大量出口狭小但是却面积大的场地，很多造成折返的障碍物，岔路。搜索轮数和f(i,j)-h(i,j)成正比，每一轮搜索，最坏情况下就是探明整个地图一遍，每一轮内没有重复劳动。
    - 最坏时间复杂度：$O((f_{i,j} - h_{i,j}) * 地图大小)$
    - 最坏空间复杂度：$O(f_{i,j} + 地图大小)$

5. 实验结果

5.1 结果和运行时间统计表

- 运行结果如下 (包括调试输出的时间)：

算法	样例	运行步数	运行时间(s)
A*	input.txt	39	0.0001673
IDA*	input.txt	39	0.0026812
A*	input2.txt	116	0.0060137
IDA*	input2.txt	116	0.0298362

其中，input.txt是助教一开始给的样例，input2.txt是助教后来补充的样例。

5.2 算法屏幕输出

- A* 算法屏幕输出如下：

1. 地图 input.txt

```
# A* start ...
find_path debug call.

# # # # # # # # # # # # # # # # # # # # # # # # #
  R #           #                               #
# D # # # #     #   #     # #   # # #   # # #   #
# D r u u # #       # #   # #   #   #   # # #   #
# D R R R # # # # # # #   # # # #   #   #   #   #
# # D # D r   #           #       # #   #   #   #
# # D # D R r #     #     #       #     #   #   #
# l D # # D R #   # # #   # # #   #         # # #
# # D R # # D #   #   #       #   #   # # # #   #
#   # # # # D #   # #   #     #   #   #   # #   #
# #   #   l D r u u #   # #   #   #   #         #
# #       l D R R R R r   #   #   #   #         #
#       # # # # D # D r # #   #   #   #     #   #
#     # #     l D # D r               #     #   #
#     #     # l D # D # #   #     #   # #   #   #
# # # #   # # # # # D # # # # u # # u u u u # u #
#         #       l D R R R R R R R R R R R R R R
# # # # # # # # # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # # # # # #
r d #           #                               #
# d # # # #     #   #     # #   # # #   # # #   #
# d       # #       # #   # #   #   #   # # #   #
# r r r d # # # # # # #   # # # #   #   #   #   #
# #   # d     #           #       # #   #   #   #
# #   # r d   #     #     #       #     #   #   #
#     # # r d #   # # #   # # #   #         # # #
# #     # # d #   #   #       #   #   # # # #   #
#   # # # # d #   # #   #     #   #   #   # #   #
# #   #     d       #   # #   #   #   #         #
# #         r r r r d     #   #   #   #         #
#       # # # #   # d   # #   #   #   #     #   #
```

```
#     # #         # d                 #     #   #
#     #     #     # d # #   #     #   # #   #   #
# # # #   # # # # # d # # # #   # #         #   #
#         #         r r r r r r r r r r r r r r
# # # # # # # # # # # # # # # # # # # # # # # # #
# check_and_showpath SUCCEED.
RDDDRRRDDRDRDDDDRRRRDDDDDRRRRRRRRRRRRRR
39

 # A* end.
```

2. 地图 input2.txt

```
# A* start ...
find_path debug call.

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
  R R R R R R R R R R R R R R R R R R R R R R R R R R R R R # U U U U U U R R R R R R # u u u u u # u u u u u u u u u #
# D R R R R R R R R R # D R R R R R R R R R R R R R R R R R # U U U U U U # D R R R R # U U U U U # U U U U U U U U U #
# D R R R R R R R R R # D R R R R R R R R R R R R R R R R R # U U U U U U # D R R R R # U U U U U # U U U U U U U U U #
# D R R R R R R R R R # D R R R R R R R R R R R R R R R R R # U U U U U U # D R R R R # U U U U U # U U U U U U U U U #
# D R R R R R R R R R # D R R R R R R R R R R R R R R R R R R R R R R R # D R R R R # U U U U U # U U U U U U U U U #
# D R R R R R R R R R # D R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R # U U U U U R R R R R R R R R R #
# D R # # # # # # # # # D R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R # U U U U U # D R R R R R R R R #
# D R # L L L L L L L L D R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R # U U U U U # D R R R R R R R R #
# D R # L L L L L L L L D R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R # U U U U U # D R R R R R R R R #
# D R # # # # # # # # # # # # # # # # # # # # # # # # # # # D R R R R R # D R R R R # U U U U U # D R R R R R R R R #
# D R R R R R R R R R R R R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R # U U U U U # D R R R R R R R R #
# D R R R R R R R R R R R R R R R R R R R R R R R R R R R R # D R R R R R # D R R R R R R R R R R # D R R R R R R R R #
# # # # # # # D # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # D #
# U U U U # L D R R R R R R # U U U U U U U U R R R R R R R R R #                           #             l L L L L D #
# U U U U # L D R R R R R R # U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# U U U U # # # # D # # # # # U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #                           #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R # # # #   # # # # # # #   # #             l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R #   u u u #               #               l L L L L D #
# L L L L L L L L D # U U U U U U U U U U U U # D R R R R R R R # u U U U #               #               l L L L L D #
# L L L L L L L L D R R R R R R R R R R R R R # D R R R R R R R # L U R R #               #               l L L L L D #
# L L L L L L L L D # D R R R R R R R R R R R # D R R R R R R R # # U # # # # # # # #     #             l L L L L L D #
# L L L L L L L L D # D R R R R R R R R R R R # D R R R R R R R R R R R R R R R R R R #   # # # # # # # # D # # # # # #
# L L L L L L L L D # D R R R R R R R R R R R # D R R R R R R R R R R R R R R R R R R #                 l D R R R R R R
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
r r r r r r r r r r r r d                                   #           r r d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # r r r r r r r r r r r r r r r r r r r r r r r r u # d         #           #                   #
#                     #                                     #             # d         #         r r d                 #
#     # # # # # # # # #                                     #             # d         #         u # d                 #
#     #                                                     #             # d         #         u # d                 #
#     #                                                     #             # d         #         u # d                 #
#     # # # # # # # # # # # # # # # # # # # # # # # # # # # #             # d         #         u # d                 #
#                                                           #             # d         #         u # d                 #
#                                                           #             # r r r r r r r r r r u # r r r r r r r r d #
# # # # # # #   # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # d #
#         #                 #                                   #                           #                       d #
#         #                 #                 #                 #                           #                       d #
#         # # # #   # # # # #                 #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 # # # #   # # # # # # #   # #                       d #
#                   #                         #                 #         #               #                         d #
#                   #                         #                 #         #               #                         d #
#                                             #                 #         #               #                         d #
#                   #                         #                 # #   # # # # # # # #     #               d l l l l l #
#                   #                         #                                     #     # # # # # # # # d # # # # # #
#                   #                         #                                     #                     r r r r r r
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# check_and_showpath SUCCEED.
RRRRRRRRRRRRDDDDRRRRRRRRRRRRRRRRRRRRRRRRRUUUURRDDDDDDDDDDDRRRRRRRRRUUUUUURRDDDDDDRRRRRRRRDDDDDDDDDDDDDDLLLLLDDRRRRRR
116

 # A* end.
```

- IDA* 算法屏幕输出如下：

1. 地图 input.txt

```
# IDA* start ...
IDAstar:  i_max:17 j_max:24
find_path: ---- path_dfs: 39 ----
dfs各层次访问次数: 1 | 2 | 1 | 1 | 1 | 2 | 2 | 4 | 3 | 3 | 4 | 3 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 4 | 3 | 3 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
 2 | 2 | 2 | 2 | 2 | 2 | 2 |
dfs总访问次数: 80

# # # # # # # # # # # # # # # # # # # # # # # # #
r d #           #                               #
# d # # # #     #   #     # #   # # #   # # #   #
# d       # #       # #   # #   #   #   # # #   #
# r r r d # # # # # # #   # # # #   #   #   #   #
# #   # d     #           #       # #   #   #   #
# #   # r d   #     #     #       #     #   #   #
#     # # r d #   # # #   # # #   #         # # #
# #     # # d #   #   #       #   #   # # # #   #
#   # # # # d #   # #   #     #   #   #   # #   #
# #   #     d       #   # #   #   #   #         #
# #         r r r r d     #   #   #   #         #
#       # # # #   # d   # #   #   #   #     #   #
#     # #         # d                 #     #   #
#     #     #     # d # #   #     #   # #   #   #
# # # #   # # # # # d # # # #   # #         #   #
#         #         r r r r r r r r r r r r r r
# # # # # # # # # # # # # # # # # # # # # # # # #
# check_and_showpath SUCCEED.
RDDDRRRDDRDRDDDDRRRRDDDDDRRRRRRRRRRRRRR
39

 # IDA* end.
```

2. 地图 input2.txt

```
# IDA* start ...
IDAstar:  i_max:29 j_max:59
find_path: ---- path_dfs: 86 ----
find_path: ---- path_dfs: 88 ----
find_path: ---- path_dfs: 90 ----
find_path: ---- path_dfs: 92 ----
find_path: ---- path_dfs: 94 ----
find_path: ---- path_dfs: 96 ----
find_path: ---- path_dfs: 98 ----
find_path: ---- path_dfs: 100 ----
find_path: ---- path_dfs: 102 ----
find_path: ---- path_dfs: 104 ----
find_path: ---- path_dfs: 106 ----
find_path: ---- path_dfs: 108 ----
find_path: ---- path_dfs: 110 ----
find_path: ---- path_dfs: 112 ----
find_path: ---- path_dfs: 114 ----
find_path: ---- path_dfs: 116 ----
dfs各层次访问次数: 1 | 2 | 2 | 6 | 10 | 14 | 18 | 22 | 26 | 30 | 30 | 30 | 30 | 25 | 21 | 17 | 13 | 10 | 14 | 18 | 26 | 34 | 46 | 46 | 42 | 46 | 46 |
50 | 50 | 46 | 46 | 50 | 58 | 66 | 74 | 78 | 70 | 74 | 74 | 78 | 82 | 83 | 83 | 83 | 79 | 70 | 62 | 57 | 49 | 41 | 37 | 33 | 29 | 25 | 21 | 17 | 13 |
 9 | 6 | 6 | 6 | 10 | 14 | 18 | 22 | 26 | 30 | 34 | 35 | 35 | 35 | 35 | 35 | 35 | 34 | 30 | 25 | 21 | 17 | 13 | 9 | 9 | 10 | 10 | 14 | 14 | 22 | 22 |
 16 | 12 | 9 | 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 4 | 4 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
dfs总访问次数: 3032

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
r r r r r r r r r r r r d                                   #           r r d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # d                                   #           u # d         #           #                   #
#                     # r r r r r r r r r r r r r r r r r r r r r r r r u # d         #           #                   #
#                     #                                     #             # d         #         r r d                 #
#     # # # # # # # # #                                     #             # d         #         u # d                 #
#     #                                                     #             # d         #         u # d                 #
#     #                                                     #             # d         #         u # d                 #
#     # # # # # # # # # # # # # # # # # # # # # # # # # # # #             # d         #         u # d                 #
#                                                           #             # d         #         u # d                 #
#                                                           #             # r r r r r r r r r r u # r r r r r r r r d #
# # # # # # #   # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # d #
#         #                 #                                   #                           #                       d #
#         #                 #                 #                 #                           #                       d #
#         # # # #   # # # # #                 #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 #                           #                       d #
#                   #                         #                 # # # #   # # # # # # #   # #                       d #
#                   #                         #                 #         #               #                         d #
#                   #                         #                 #         #               #                         d #
#                                             #                 #         #               #                         d #
#                   #                         #                 # #   # # # # # # # #     #               d l l l l l #
```

```
#                   #                         #                                     #     # # # # # # # # d # # # # # #
#                   #                         #                                     #                     r r r r r r
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
# check_and_showpath SUCCEED.
RRRRRRRRRRRRDDDDRRRRRRRRRRRRRRRRRRRRRRRRUUUURRDDDDDDDDDDDRRRRRRRRRRUUUUUURRDDDDDDRRRRRRRRDDDDDDDDDDDDDDLLLLLDDRRRRRR
116

 # IDA* end.
```

- 这两个算法输出到output_A.txt和output_IDA.txt的文件内容略，详见doc文件夹下备份的文件。

6. 实验总结

通过本此实验，我对A*算法加深了理解，也和IDA * 算法有了对比，对他们的算法性能，优化办法和实现方式都有了更深入的理解。