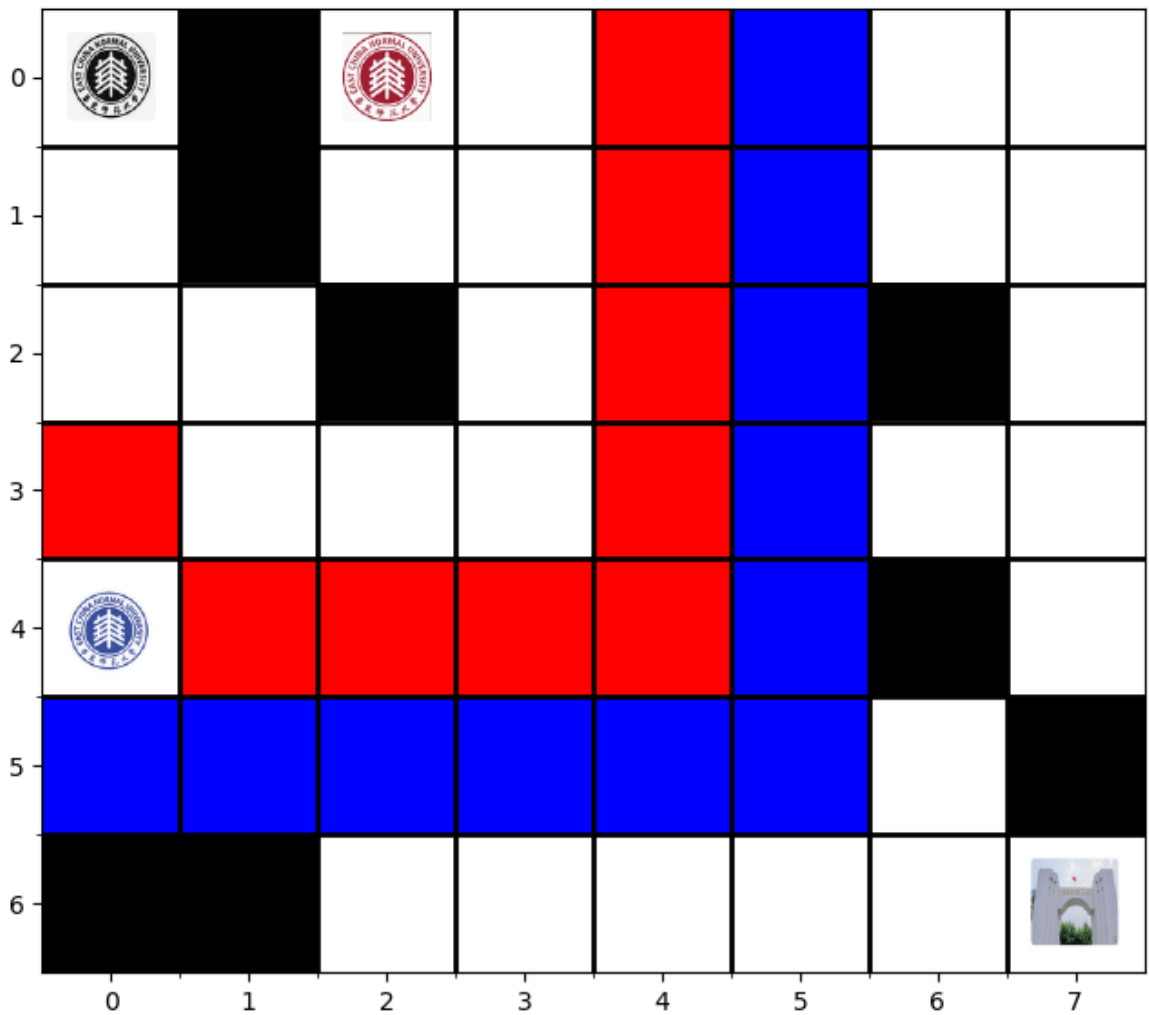


# dfs,bfs,dijkstra,a\*求解迷宫问题

## 运行要求

- Python
- numpy
- matplotlib
- 绘制迷宫所需的图片以正确的文件名保存在正确的位置

# 迷宫描述



迷宫是一个二维数组，每个格子都有上下左右四个方向，通过上下左右四个方向可以到达相邻的格子。华东师范大学黑色校徽需要从迷宫左上角到迷宫右下角（华东师范大学校门）。迷宫有三种障碍，分别为黑色，蓝色，红色。黑色障碍永远不能逾越。红色障碍在经过华东师范大学的红色校徽后，可以逾越。蓝色障碍在经过华东师范大学的蓝色校徽后，可以逾越。

## 代码思路

由于是否经过红色和蓝色校徽会影响整个迷宫的形状，所以将每个点看成一个四维的点  $node_{statu1,statu2,x,y}$   $statu1,statu2$  的取值为 0 或 1 表示是否经过红色或蓝色校徽， $x,y$  表示当前处于  $(x,y)$ 。按照此方式建图，注意在经过校徽时更新  $statu1,statu2$  以及在不同状态下不同种类的障碍。剩余的最短路求法分别采用  $bfs,dfs,dijkstra,A^*$  完成。该算法还可以扩展至墙的种类更多的

情况，用  $dis_{sta,x,y}$  表示，如果第  $i$  种墙的逾越条件能够满足，那么  $sta$  的二进制下面的第  $i$  位为1，否则为 0。

对于dfs,为了防止走重复的位置，采用了记忆化搜索的方式

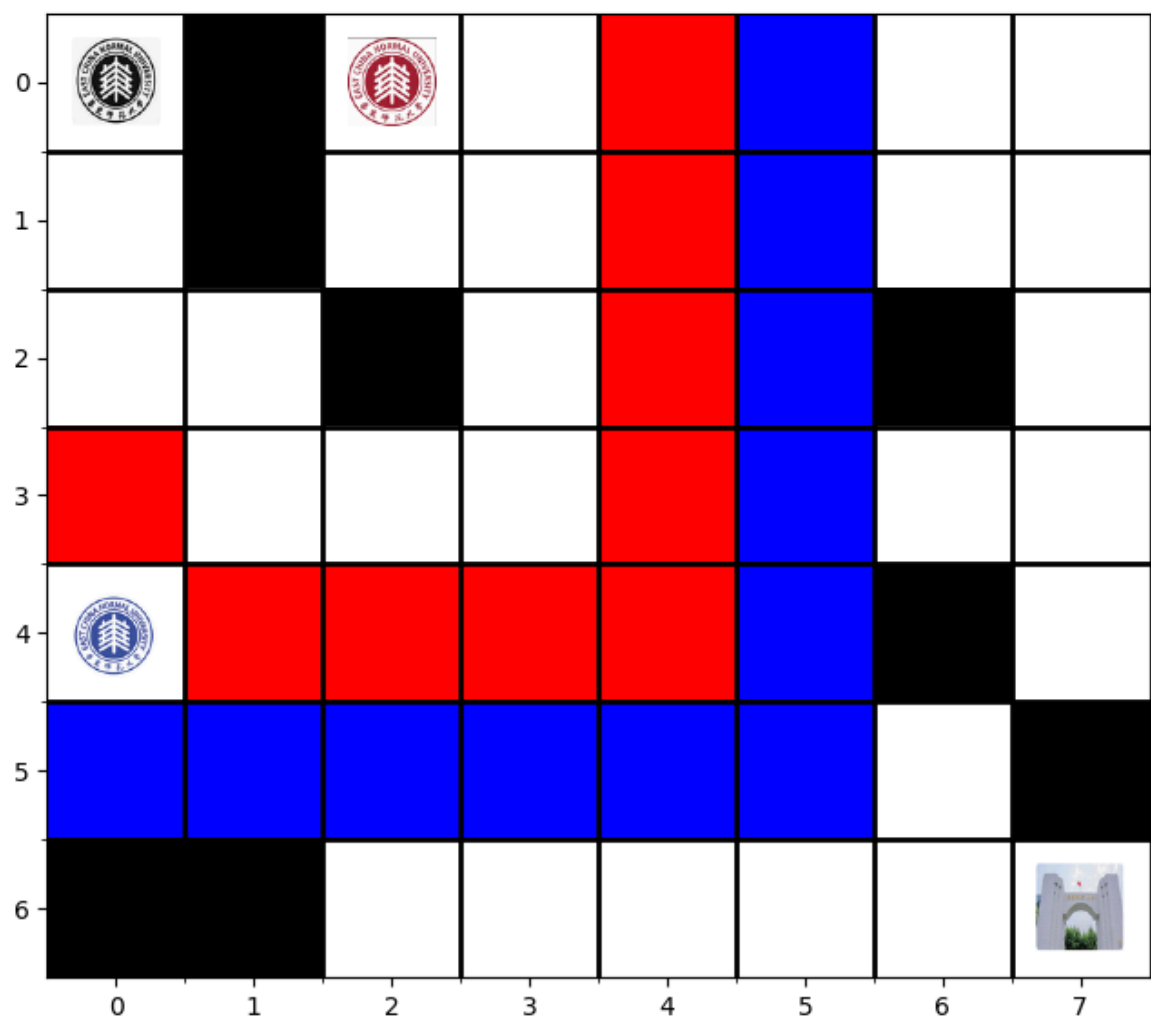
```
if dis[t1][t2][x][y]<=len(path):  
    return  
dis[t1][t2][x][y]=len(path)
```

对于  $A^*$ ，采用了到终点的曼哈段距离作为评估函数

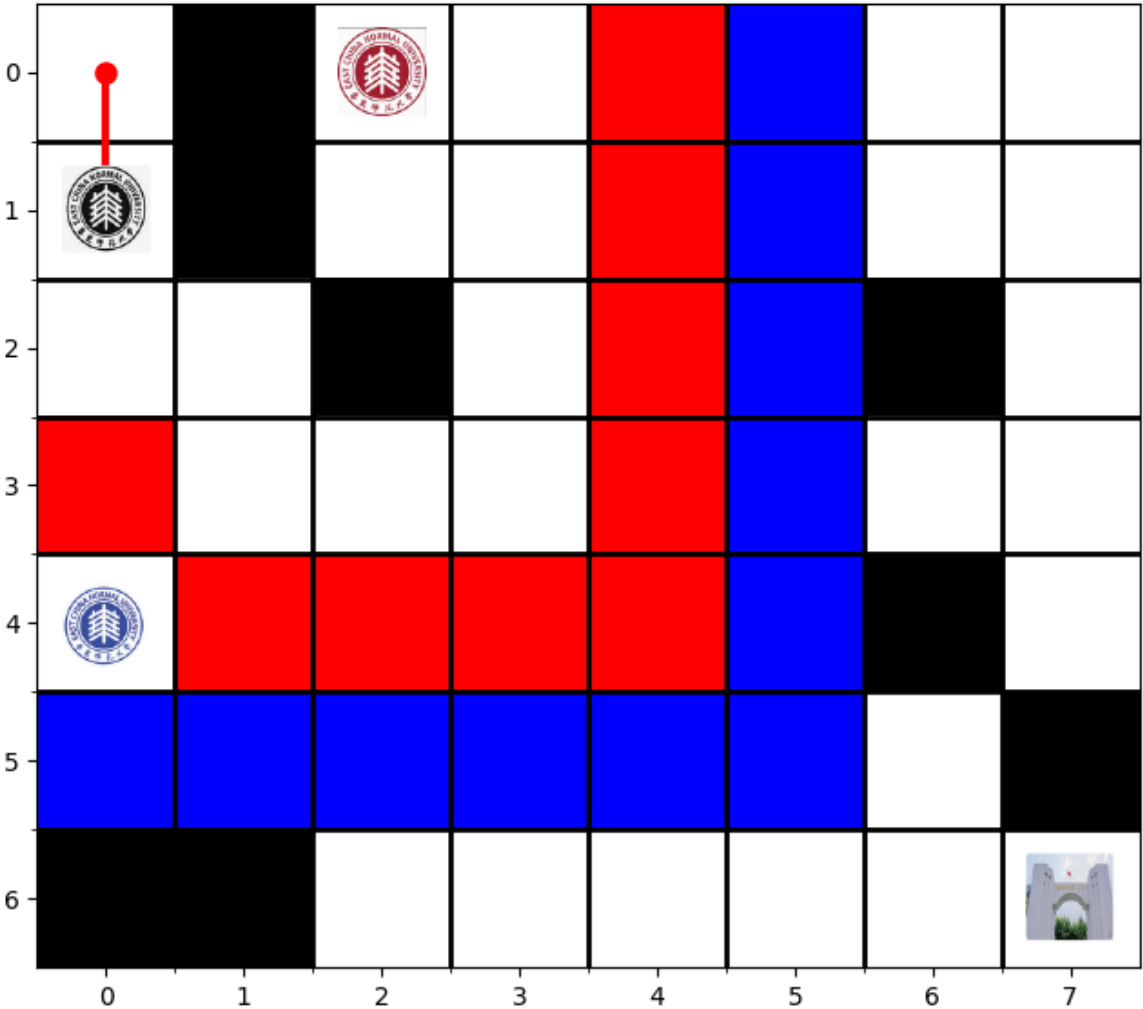
## 效果展示

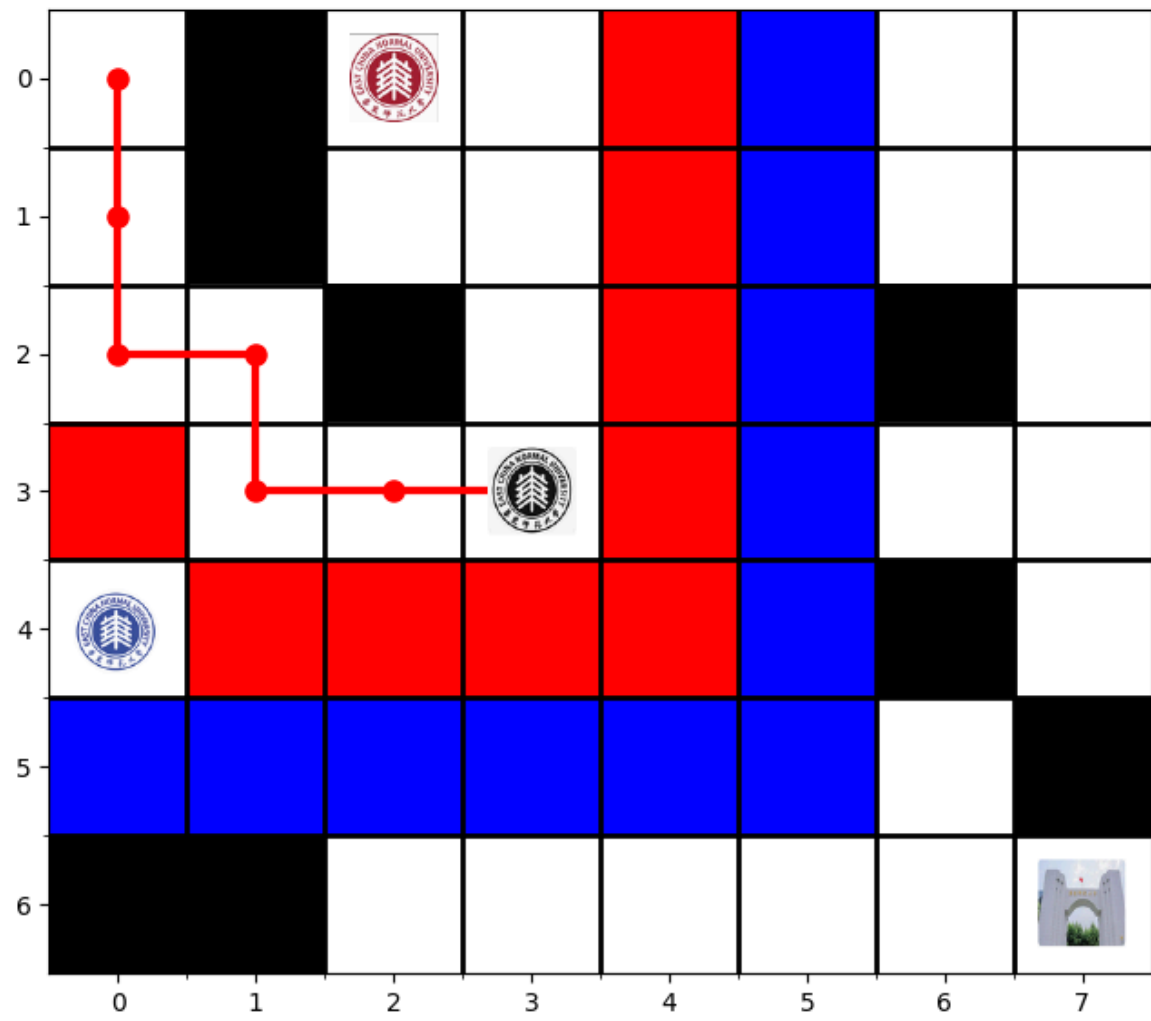
这里选出了  $A^*$  算法求解过程中的部分状态

初始状态

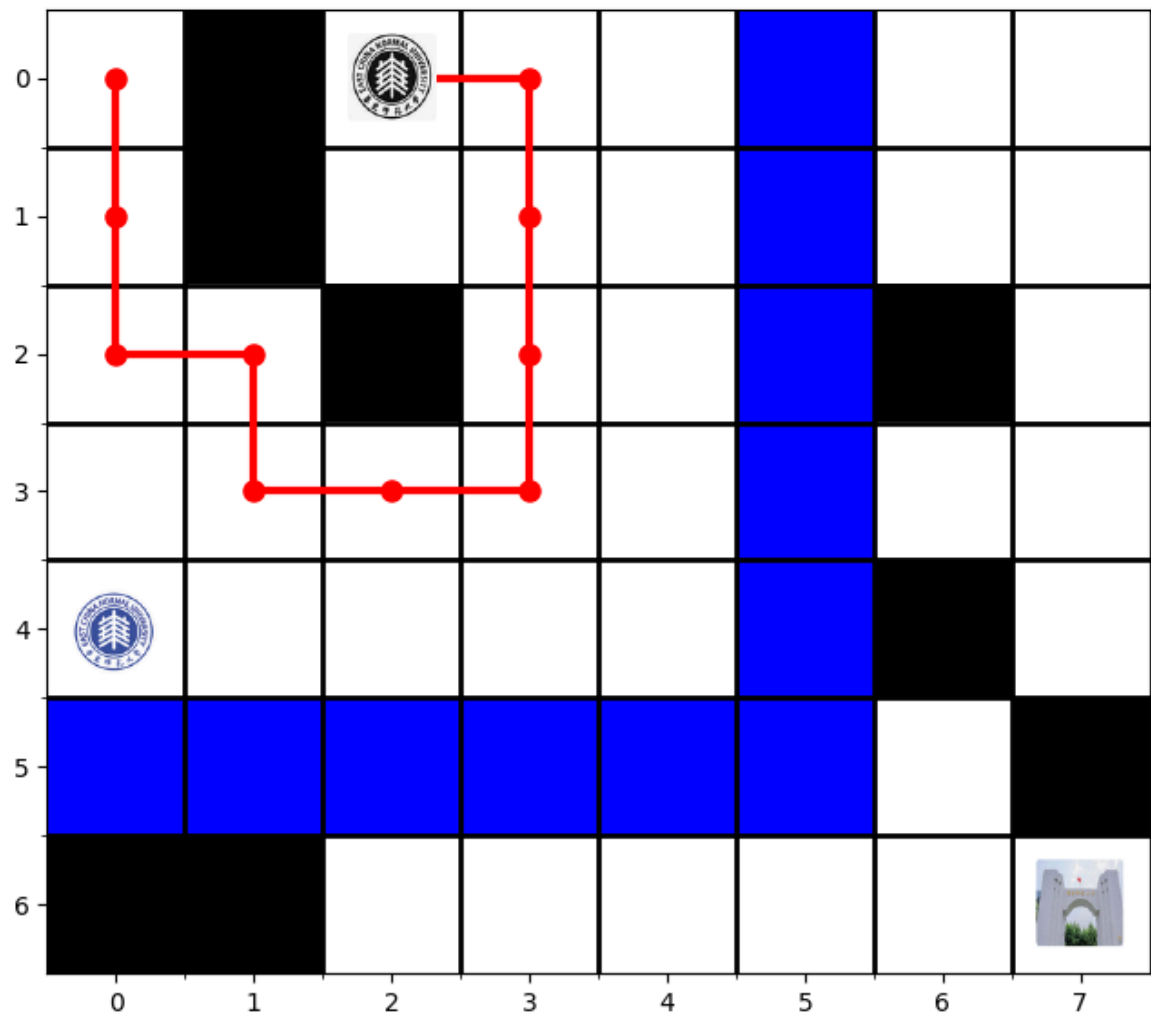


# 走向红色校徽

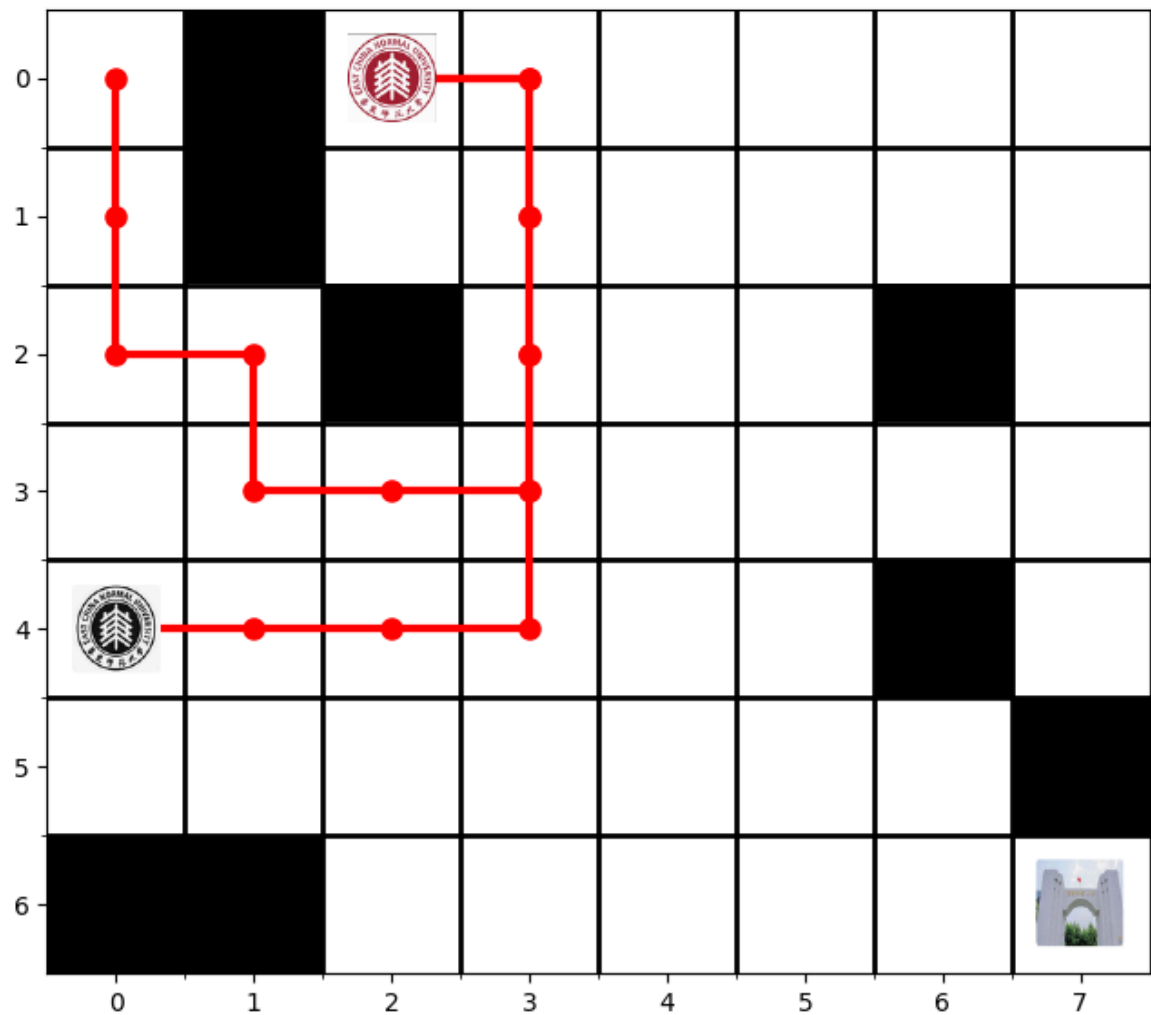




到达红色校徽，红色的障碍消失



到达了蓝色校徽，蓝色的障碍消失





到达终点

