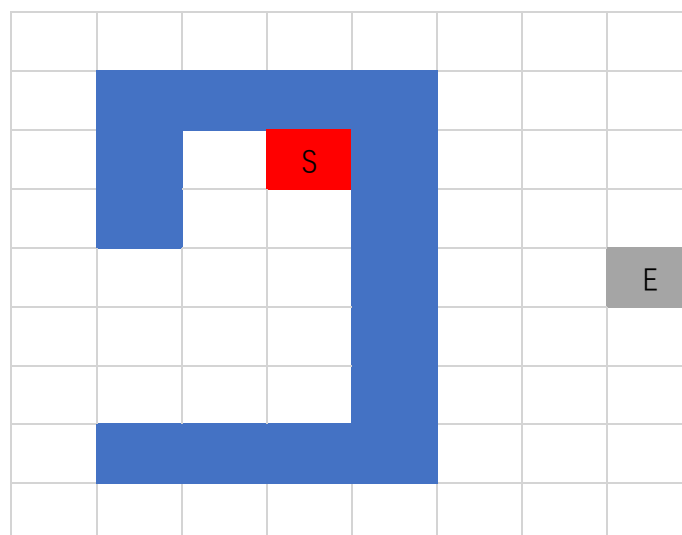


2. 搜索模块：机器人避障寻径问题

2.1 实验内容

机器人避障寻径是机器人领域的一项重要技术，该技术要求实现机器人在复杂环境中的自主导航。在现实生活中，机器人的自主导航需要考虑各种因素，包括环境的复杂性、障碍物的位置和形状、机器人的运动能力等。如何解决这些问题是机器人避障寻径任务的研究难点。基于规则的传统路径规划方法，如 A* 算法、Dijkstra 算法等，已经在机器人路径规划中得到广泛应用，并能够在大部分场景中实现较高的准确度。这些算法的核心思想是通过搜索算法在地图中找到最优的路径。这些算法具有计算复杂度低、实现简单等优点。

在本实验中，我们给定如下的机器人避障寻径场景图：



其中 S 为机器人寻径的起点，E 为寻径的终点，蓝色区域为无法通过的障碍。要求机器人从 S 点出发，绕开障碍，到达 E 点。本实验需要参与者自行设计核心算法规则完成机器人从起点到终点的最短路径搜索。

2.2 实验要求

- 1) 已提供场景图的的矩阵表示，以及机器人起始点和终止点的坐标
- 2) 要求找到从起始点到终止点的最短路径
- 3) 禁止采用 hard code 的方式
- 4) 可视化最短路径图搜索过程

2.3 选做内容(加分项)

1) 自主设计更高难度场景图，并结合强化学习、深度强化学习等算法解决问题

2) 分析所设计算法的局限性，及改进点

3) 最短路径查找过程的动态化展示

4) 提出第二种完全不同的解决方案

2.4 实验环境

建议采用 python 来实现目标算法。允许调用基于 Python 的 Pandas、Numpy、Sklearn 等算法库。Pytorch 等深度学习框架也是被鼓励使用的。

2.5 注意事项

在做选做过程中，如果用到深度学习模型，需要搭建模型运行环境。为保证深度学习模型的正常运行，不同 python 库之间版本需要正确匹配。建议使用 anaconda 自动管理各种 python 库。

2.6 参考资料

Python:

<https://www.runoob.com/python3/python3-tutorial.html>

Numpy:

<https://www.numpy.org>

Sklearn:

<https://scikit-learn.org>

Pytorch:

<https://pytorch.org/>

参考文献:

A note on two problems in connexion with graphs