

# 数据结构 Project：智能仓储机器人路径规划

## 数据结构

2025 年秋季学期

### 1 简要信息

**项目名称：**智能仓储机器人 (Smart Warehouse AGV)

**背景描述：**在现代化的电商物流中心，成千上万的 AGV (自动导引车) 在货架间穿梭。它们需要从 **充电站 (S)** 出发，前往指定的 **货架 (T)** 取货，最后将货物运送到 **打包台 (E)**。本项目的任务是设计并实现一个程序，为机器人规划一条满足作业顺序且不发生碰撞的最短路径。

此项目涵盖了图的遍历 (BFS)、队列等基本数据结构的实际应用。

**提交截止日期：** 2025.12.28 23:59

### 2 Project 要求

#### 2.1 基本要求

##### 1. 实现基础数据结构

- 设计存储仓库地图的数据结构（建议使用二维数组或向量）。
- 设计记录机器人位置和路径的数据结构。

##### 2. 完成多阶段路径搜索功能

- 机器人必须严格遵守作业顺序：**起点 (S) → 货物点 (T) → 终点 (E)**。
- 直接从起点 (S) 到达终点 (E) 被视为**任务未完成**。
- 程序需输出完整的坐标序列；若无法到达货物点，或从货物点无法到达终点，则输出“无解”。

#### 2.2 高级要求

##### 1. 电量（步数）预警检测

- 输入数据将提供机器人的最大电量（支持的最大移动步数  $K$ ）。
- 如果规划出的总路径长度（ $S \rightarrow T$  的步数 +  $T \rightarrow E$  的步数）超过  $K$ ，即使路径存在，也需输出“电量不足”。

## 2. 可视化地图和路径

- 在终端打印地图，并用特定字符（如 \* 或 .）标记出规划好的路径，直观展示机器人的移动轨迹。用 . 标记第一阶段路径（去取货），用 \* 标记第二阶段路径（去送货）。

# 3 实现细节

## 3.1 输入格式

输入第一行包含 3 个正整数，分别为地图的行数  $M$ 、列数  $N$  ( $1 \leq M, N \leq 50$ ) 以及机器人的最大电量步数  $K$ 。

随后输入  $M$  行，每行  $N$  个字符，代表地图具体信息：

- S：起点 (Start/充电站)
- E：终点 (End/打包台)
- T：货物 (Target/货架)
- 0：可通行空地
- 1：不可通行的障碍物

## 3.2 输出格式

- 若存在满足电量限制的路径，按顺序输出经过的坐标序列，格式如  $(r, c)$ ，并输出总步数。
- 若路径不存在（被围住）或总步数超过  $K$ ，输出对应的错误提示（如“无解”或“电量不足”）。

## 3.3 样例说明

### 测试样例 1 (正常任务)

输入：

```
1 5 5 20
2 S0100
3 00100
4 00T00
5 11100
6 0000E
```

输出 (示例):

```
1 (0, 0) -> (1, 0) -> (2, 0) -> (2, 1) -> (2, 2) -> (2, 3) -> (3, 3) ->
  (4, 3) -> (4, 4)
2 总步数: 8
3 状态: 任务完成
4
5 可视化路径:
6 S0100
7 .0100
8 ..T*0
9 111*0
10 000*E
```

## 测试样例 2 (电量不足)

输入:

```
1 3 3 2
2 S00
3 0T0
4 00E
```

解释: 从  $S$  到  $T$  最少需要 2 步, 从  $T$  到  $E$  最少需要 2 步, 总共 4 步。但电量  $K=2$ 。

输出:

```
1 电量不足
```

## 4 提交要求

作业通过 eLearning 进行提交。提交文件应为一个文件名为 **学号**, 后缀名为 **.zip** 的压缩包。结构如下:

```
1 StudentId.zip
2 |__ report.pdf # 项目报告
3 |__ main.cpp # 核心源代码
```

**report.pdf 要求：**长度不宜超过 10 页。内容应包括：

1. 解题思路：如何处理分段路径搜索 ( $S \rightarrow T$  和  $T \rightarrow E$ )。
2. 测试结果：展示你的程序运行样例截图。
3. 遇到的问题与收获。

**main.cpp 要求：**

1. 需要包含必要的中文注释。
2. 请使用 UTF-8 编码提交。

**学术诚信：**若参考了互联网资料或获得同学帮助，请在报告末尾注明。严禁直接抄袭代码，一经发现将酌情扣分或记零分。

## 5 项目评分

### 5.1 评分组成

本项目评分包括代码评分（60%）和报告评分（40%）。

### 5.2 代码评分环境

代码评分将在 Linux 环境下（如 Ubuntu 22.04）进行编译。

- **C++ 提交：**建议将代码放在单个 main.cpp 文件中。编译命令参考：

```
1 g++ -Wall -std=c++20 -O2 main.cpp -o main
```

- **禁止事项：**禁止使用任何第三方库（如 Boost 等），仅允许使用 C++ 标准库 (STL)。

### 5.3 文档评分

在代码能正确运行的前提下，报告逻辑清晰、能准确说明设计思路即可获得全部分数。请勿堆砌大量代码，重在逻辑阐述。