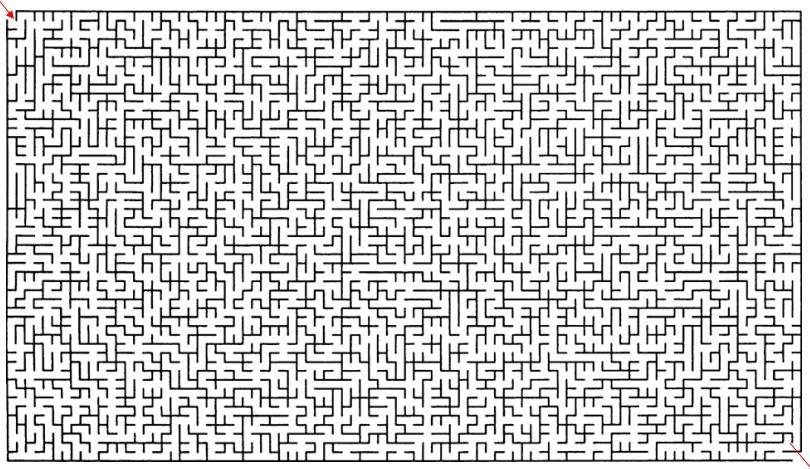
**实验一**

1. **实验名称：**迷宫
2. **需求和规格说明：**

**2.1引言：**

使用 Union-Find 数据结构完成一个迷宫的生成。迷宫的入口点位于左上角，出口点是在图的右下角

如该图所示的迷宫是由 50\*88 个单元组成的矩形，在该矩形中，左上角的单元被连通到右下角的单元，而且这些单元与相邻的单元通过墙壁分离开来。

**2.2需求概述：**

1.输入：

输入两个整数，代表所生成迷宫对应矩阵的大小

2.输出：

1）用合理的方式展现生成的迷宫；

2）给出迷宫可以走通的路径。以 SEN…（代表向南，然后向东，然后再向北，等等） 的形式给出输出结果。

**2.3接口需求：**

无

**2.4需求分析：**

该迷宫需要实现的功能有能按用户输入的两个整数表示任意m\*n大小迷宫，迷宫之间用墙隔开，并且还要用指定的方向符号表示出可从入口走到出口的一组答案。

3. 设计

3.1 设计思想：

（1）迷宫表示：由于用m\*n矩阵表示迷宫的大小

（2）迷宫路径初始化：使用Union-Find数据结构开辟迷宫路径

（3）查找路径：用深度优先搜索DFS查找一条通路

（4）打印路径：用N、E、W、S打印路径

3.2 设计表示：

创建Functions接口，包含以下功能的函数

|  |  |
| --- | --- |
| 初始化迷宫 | 随机生成迷宫 |
| 解决迷宫 | 初始化并查集 |
| 把二维数组“房间”下标转为适合并查集的下标 | 并查集：查找集合 |
| 并查集：合并集合 | 判断两个集合（元素、点）是否连通 |
| 打印迷宫 | 打印迷宫路径 |
| 恢复迷宫初始状态 | 深度优先查找迷宫路径 |

3.3 实现注释：各项功能的实现程度、在完成基本要求的基础上还实现了什么功能；

实现了项目要求的

1. 输入两个整数创建m\*n大小迷宫
2. 合理方式表示迷宫。用‘+’表示墙角，用‘-’和‘|’表示墙壁,‘ ’表示通路
3. 给出迷宫可以走通的路径。用N、S、W、E表示了迷宫可通过的路径。

此外还额外实现了在迷宫之内用‘#’表示迷宫可走的一条路径。

3.4 详细设计表示:主要算法的框架；

（1）迷宫表示：由于用m\*n矩阵表示迷宫的大小，定义一个二维字符型数组char[][] maze，但是由于设计思想是为每个矩阵方格用墙角符号‘+’和墙壁符号‘-’或‘|’来表示，所以二维数组的行数和列数为2m+1和2n+1；所以定义函数Maze(**int** pRow, **int** pColumn)

有row = 2\*pRow + 1;且column = 2\*pColumn + 1;

（2）迷宫路径初始化：用二维字符数组初始化迷宫完成以后，此时的迷宫除入口和出口以外是全封闭的。使用Union-Find数据结构开辟迷宫路径，设计思想为从出口开始随机查找一面为‘-’或‘|’的墙，如果该墙所连接的迷宫方格之间不连通，则拆掉这面墙，反之如果连通查找下一面墙，直至查找至出口，这样就保证了迷宫至少有一条通路。

（3）查找打印路径：用深度优先搜索DFS查找一条通路，并用符号‘#’表示路径的走法，最后用N、E、W、S打印路径

4. 调试报告：调试过程中遇到的主要问题是如何解决的；对于设计和编码的回顾讨论 和分析；时空分析；改进设想；

5. 运行结果展示：通过截图的方式全方位展示程序的运行结果。