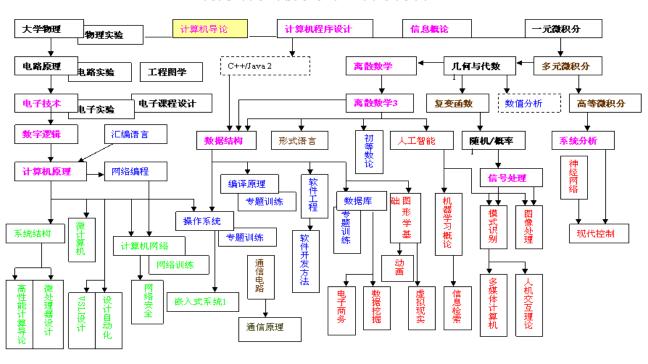


数据结构与算法



计算机科学与技术系专业课程关系图





《数据结构与算法》内容研究

- 为什么要学习数据结构与算法?
- 数据结构主要内容
- 算法及分析方法





思考

我们需要计算机来做什么?

如何让计算机做?

程序=算法+数据结构,而算法=逻辑+控制



程序=算法+数据结构,而算法=逻辑+控制





需要解决以下几个问题:

- → 输入?
 - 迷宫地图
 - 入口与出口
- → 輸出?
 - 入口到出口的路径
- 輸入如何转換为輸出?
 - 在已知迷宫地图寻找入口到出口路径的方法(算法)



如何表示给定的空间和可行的路径? 如何表示入口和出口?

寻路过程:

当有多条可行的路径时如何选择?

当某条路径在某一点再没有可行之路时如何处理理?





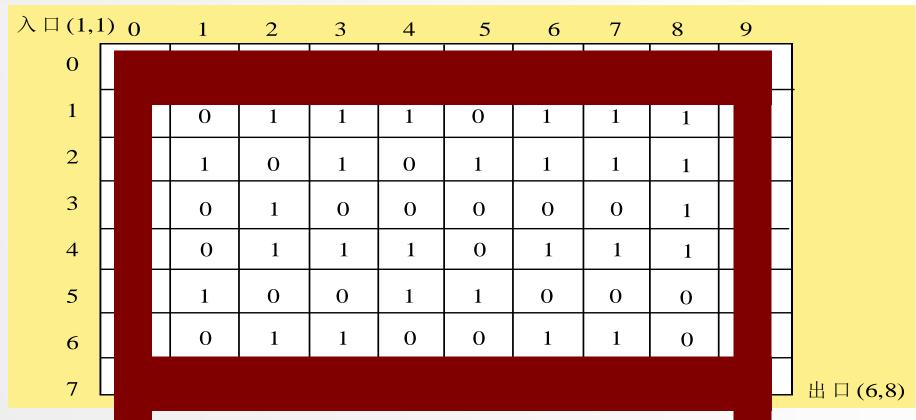
2. 迷宫表示

可以用一个m行n列的二维数组maze[m][n]来表示迷宫空间(或称迷宫地图), 并约定:

当数组元素maze[i][j]=0,表示通路, maze[i][j]=1,表示不通;

◎数据结构与算法 | Data Structures and Algorithms





入口为左上角maze[1][1],出口为右下角maze[m]<mark>[n]</mark>;

Back



如何表示给定的空间和可行的路径?

如何表示入口和出口?

寻路过程:

当有多条可行的路径时如何选择?

当某条路径在某一点再没有可行之路时如何处理理?





多条可行路

在某一点(x, y), 有8个可以探索的方向:

x-1, y-1	x, y-1	x+1, y-1
x-1, y	х, у	x+1, y
x-1, y+1	x, y+1	x+1, y+1

假设: 从正东方向开始,沿顺时针方向依次进行探索



探索方向存储: 增量数组DeltaXY

Problem:角点,边点和中间点探测 判断方法是一致的吗?

增量数组DeltaXY

X	У
1	0
1	1
0	1
-1	1
-1	0
-1	-1
0	-1
1	-1



角点、边点与中点探测方法一致性处理

----迷宫地图简化

为了使探索方向个数一致,可在原来的迷宫地图四周围都扩展一个点,即增加两行和两列,并将迷宫四周增加点的值全部置为1,表示是墙,不能通行。

这样做使得原迷宫地图中的所有点都成为了中间点,不用再判断当前点是 角点、边点、还是中间点,每个点的探索方向均为8个。

◎数据结构与算法 | Data Structures and Algorithms



入口(1,1	1) 0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1
2	1	1	0	1	0	1	1	1	1	1
3	1	0	1	0	0	0	0	0	1	1
4	1	0	1	1	1	0	1	1	1	1
5	1	1	0	0	1	1	0	0	0	1
6	1	0	1	1	0	0	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1

入口为左上角maze[1][1], 出口为右下角maze[m][n];





如何表示给定的空间和可行的路径?

如何表示入口和出口?

寻路过程:

当有多条可行的路径时如何选择?

当某条路径在某一点再没有可行之路时如何处理理?





死路退回:

回退到上一个具有多条路径的地方选择下一条路径探测;

需要存储每一个具有多条路径的点坐标和探索方向(x,y,d),

存储的多个(x,y,d)选择最新存储位置回退

- - 桟





如何表示给定的空间和可行的路径?

如何表示入口和出口?

寻路过程:

当有多条可行的路径时如何选择?

当某条路径在某一点再没有可行之路时如何处理理?





兜死圈子

解决兜死圈子,有两种方法:

设置一个标志数组mark[m][n],所有元素初始化为0;

在探索中,当所探索的点(i, j)对应的mark[i][j]=0时,才进入该点,并将mark[i][j]置为1;

当所探索的点(i, j)对应的mark[i][j]=1时,表明已探索过,不需要再进入。

2. 当到达某点(i, j)后,在迷宫地图的该点坐标上标记特殊值,例如将 maze[i][j]置-1,以便区别未到达过的点。



如何表示给定的空间和可行的路径?

如何表示入口和出口?

寻路过程:

当有多条可行的路径时如何选择?

当某条路径在某一点再没有可行之路时如何处理理?

某点重复经过吗(避免兜圈子)?

前两个问题属于数据结构选择和设计,后三个问题涉及算法设计。





小结:

- 1. 二维数组maze[m+2][n+2]来表示迷宫,解决了迷宫地图的存储;
- 2. 一维数组DeltaXY[8]来记载了8个探索方向的坐标增量,将8个探索方向数字 化为0到7,并将向下一点前进的操作统一为当前点的坐标+沿该探索方向的增量, 即可得到下一点的坐标;
- 3. 当某点无路可通行时,需要从该点返回到前一点,再从前一点选择下一个方向继续进行探索,即需要知道前一点和前一点当前探索的方向。因此,我们需要保留依次到达的各点的坐标和到达该点的方向;
- 4. 还需要防止重复到达某点,避免在迷宫中兜死圈子,需要记载已到达过的点。



迷宫小结

数据结构有两大用途:

一是用于存放要处理的数据,如迷宫地图;

二是用于实现算法策略,如迷宫例子中探索方向增量数

组、回溯的栈、避免重复走的标志数组或特殊标记)