万能的解题金钥匙——搜索

基础概念

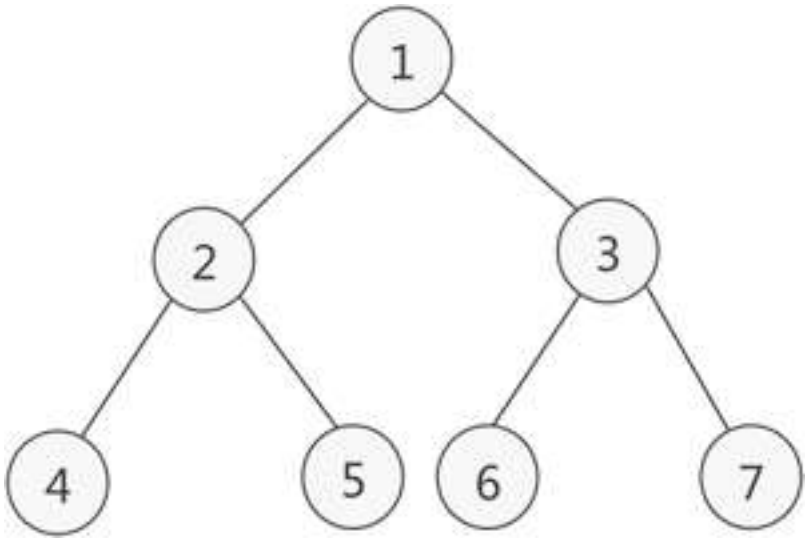
- 在我们遇到的一些问题当中，有些问题不能够确切的建立数学模型，或即便有数学模型但该模型的准确方法也不一定能运用现成算法。在要求枚举方案时，常常会遇到这一类问题。
- 解决这一类问题，我们一般采用搜索的方法解决，即从初始状态出发，运用题目所给出的条件和规则扩展所有可能情况，从中找出满足题意要求的解答。
- 状态：指当前所面临的具体问题
- 转移：指从一个状态到另一状态的一种决策
- 状态和转移可能是题目中已经给出，也可能是需要自己分析出的。一道题的状态与决策可能有多种。

BFS-广度优先搜索

Breadth-First Search

- 在搜索算法中，我们对每个节点进行拓展，深度越小的结点越先得到扩展，就是广度优先搜索法。**简而言之就是一层一层扩展，由近到远。通常解决最小花费，最短时间之类的问题。**

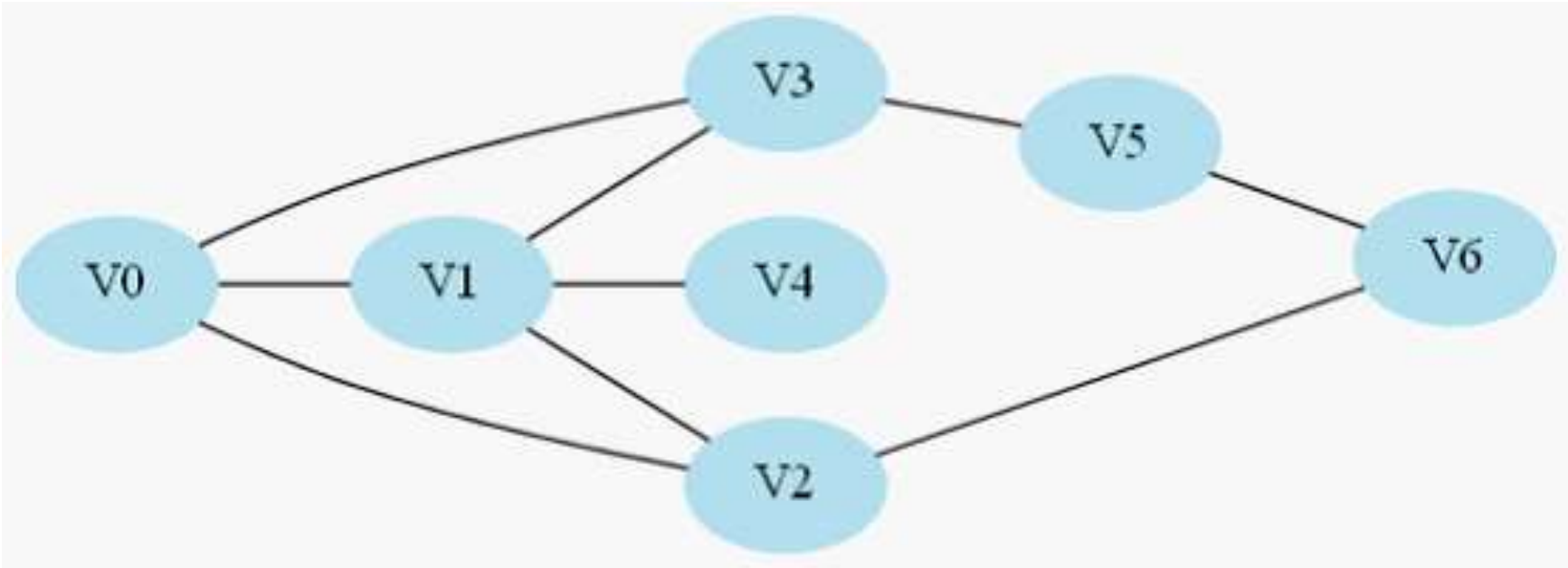
来个栗子



bfs：1 2 3 4 5 6 7 一层一层来

从这张图我们可以看出为什么bfs可以求
最小花费，最短时间
因为每条边的边权都是1
从一个状态到另一个状态的代价是相同的
所以近的会先搜到——也就是最短的会先搜到

这就是bfs能求最小花费，最短时间的原因



从V0开始搜，可以搜到V1 V2 V3
从V1开始搜也可以搜到V2 V3

第一次搜到的是最短的

算法步骤

- 1. 首先将**根节点**放入**队列**中。
- 2. 从队列中取出**第一个节点**，并检验它是否为目标。如果找到目标，则结束搜寻并回传结果。否则将它所有尚未检验过的直接子节点加入队列中。（也可以在搜索到的时候检验，看个人习惯）
- 3. **若队列为空，表示整张图都检查过了**——亦即图中没有欲搜寻的目标。结束搜寻并回传“找不到目标”。
- 4. 重复步骤2。

- 主要是学代码。

```
int bfs()
{
    queue<node>q;//node是某种数据类型(可以是自己定义的数据类型)
    q.push(起点);//把起点放入队列
    while(!q.empty())//如果队列不空 说明没有搜完
    {
        node t=q.front();//每次取出头节点
        q.pop();//弹出头节点
        //也可以在取出队头的时候判断是否得到答案。
        //对头节点进行搜索：这里假设node是一个坐标
        for(int i=0;i<4;i++)//搜索方式 这里是上下左右地搜索
        {
            int xx=t.x+dx[i],yy=t.y+dy[i];//搜到的新状态
            if(新状态==目标状态)
                return 答案;//搜到了就返回题目的答案,也可以不在这里判断是否搜到

            if(满足某些条件)//这里把符合条件的状态加入队列中
                q.push(新状态);
        }
    }
    return 题目要求的答案;
}
```

模板题:844. 走迷宫 - AcWing题库 （也是**最短路**）

- （数据范围1<=n、m<=100）

给定一个 $n \times m$ 的二维整数数组，用来表示一个迷宫，数组中只包含 0 或 1，其中 0 表示可以走的路，1 表示不可通过的墙壁。

最初，有一个人位于左上角 $(1,1)$ 处，已知该人每次可以向上、下、左、右任意一个方向移动一个位置。

请问，该人从左上角移动至右下角 (n,m) 处，至少需要移动多少次。

数据保证 $(1,1)$ 处和 (n,m) 处的数字为 0，且一定至少存在一条通路。

输入格式

第一行包含两个整数 n 和 m。

接下来 n 行，每行包含 m 个整数（0 或 1），表示完整的二维数组迷宫。

输出格式

输出一个整数，表示从左上角移动至右下角的最少移动次数。

从(1,1)到(5,5) 见图4、5

```
0 1 0 0 0
0 1 0 1 0
0 0 0 0 0
0 1 1 1 0
0 0 0 1 0
```

一些例题：P2802 回家

- P2802 回家 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)

P2802 回家 题解

- 一个简单的bfs:
- 要注意如何标记v[N][N]数据:
- **要用到(i,j)点的hp标记v[i][j]**，而不是简单地用v[i][j]=1来表示是否来过。（反例：若用v[i][j]=1来标记是否来过，假设第一次来的时候hp为1，v[i][j]=1后，由于血量不够不会往(I,j)的四周搜索，若之后有hp=6的再次来到(i,j)点，由于v[i][j]==1,就不会再搜，则(i，j)周围的点没有搜索过）
- 用hp标记v数组，但凡出现当前状态的hp>v标记的值，使该节点入队，并更新v数组。

例题2：3984 -- 迷宫问题 (poj.org)

定义一个二维数组：

```
int maze[5][5] = {
0, 1, 0, 0, 0,
0, 1, 0, 1, 0,
0, 0, 0, 0, 0,
0, 1, 1, 1, 0,
0, 0, 0, 1, 0,
};
```

它表示一个迷宫，其中的**1表示墙壁，0表示可以走的路**，
只能**横着走或竖着走**，不能斜着走，要求编程序找出：
从**左上角到右下角**的**最短路线**。

注意poj不能用万能头

Sample Input

```
0 1 0 0 0
0 1 0 1 0
0 0 0 0 0
0 1 1 1 0
0 0 0 1 0
```

Sample Output

```
(0, 0)
(1, 0)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(2, 4)
(3, 4)
(4, 4)
```

迷宫问题题解

- Bfs每个节点的状态：当前坐标，转移成当前坐标的上一个坐标。
- 注意输出格式。

例题：P1162 填涂颜色

- P1162 填涂颜色 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)

P1162 填涂颜色 题解

- 由于闭合圈内的空间是连通的，所以只要找到闭合圈内的第一个点，对它进行bfs即可。

总的代码

- [bfs题目代码_karshey的博客-CSDN博客]