

声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

《L4-CSP-J 复赛进阶算法与数据结构》

Day3 广度优先搜索BFS

主讲人：小武老师

广度优先搜索

BFS 本质就是让你在一幅「图」中找到从起点 start 到终点 target 的最近距离。对的，就是这么朴实无华且枯燥。



关于搜索



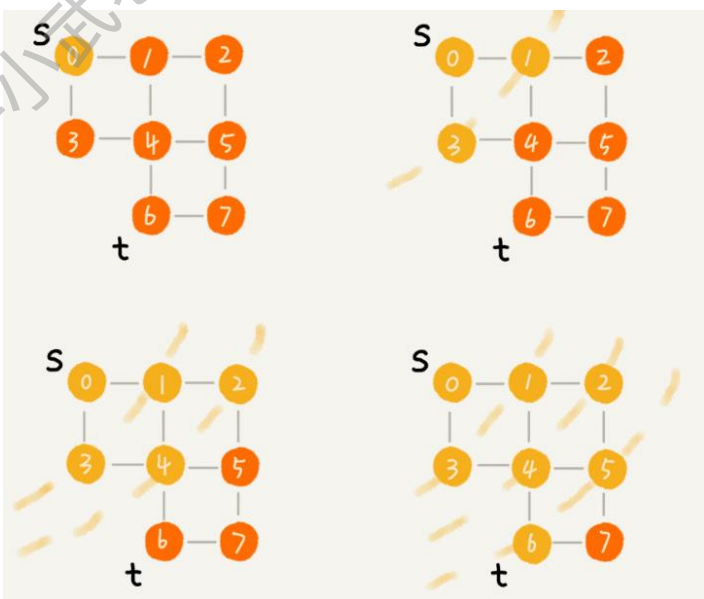
万能的解题金钥匙——搜索



广度优先搜索



广度优先搜索（Breadth First Search, BFS）。直观上来说，它其实是一种“地毯式”层层推进的搜索策略。通过BFS搜索找到的源点到终点的路径也是顶点s到顶点t的最短路径。

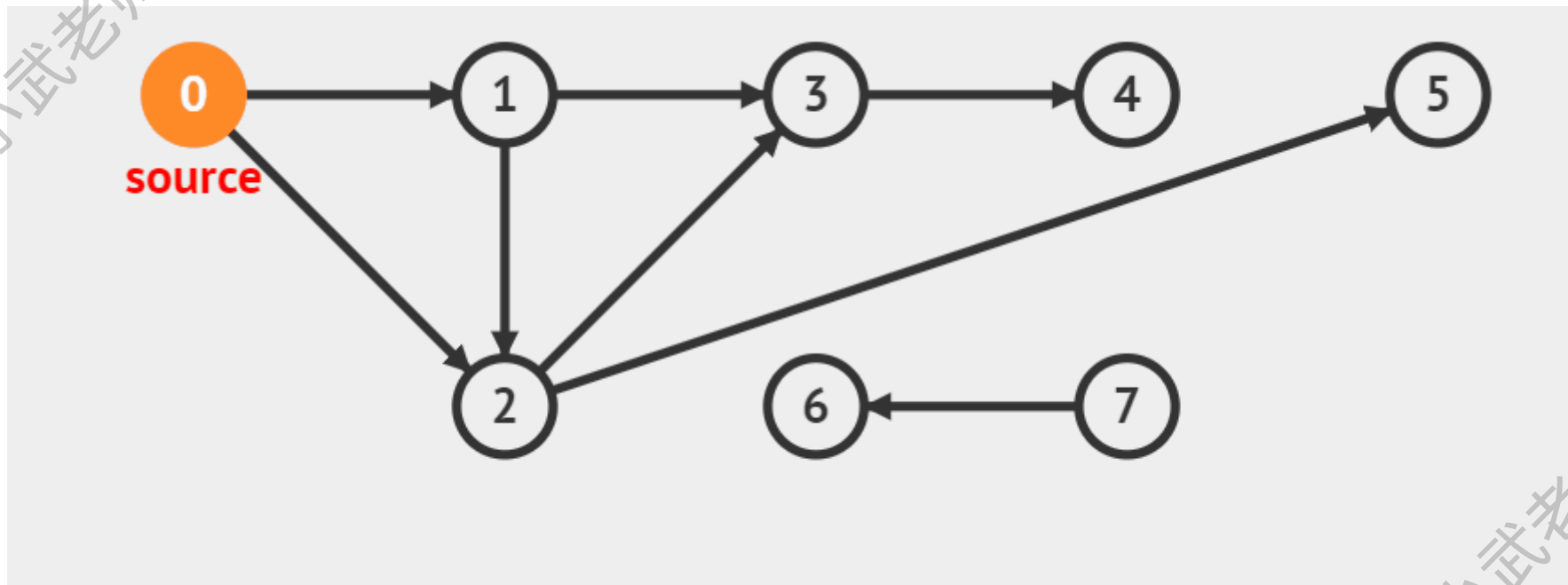


算法过程可以看做是图上火苗传播的过程：最开始只有起点着火了，在每一时刻，有火的节点都向它相邻的所有节点传播火苗

算法是作用于具体数据结构之上的，大部分搜索算法是基于“图”这种数据结构。因为图的表达能力极强，大部分涉及搜索的场景可以抽象成“图”。



广度优先搜索



<https://visualgo.net>

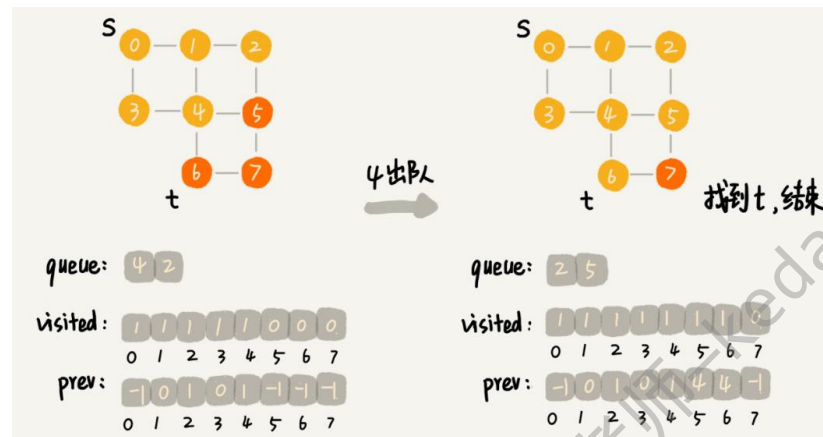
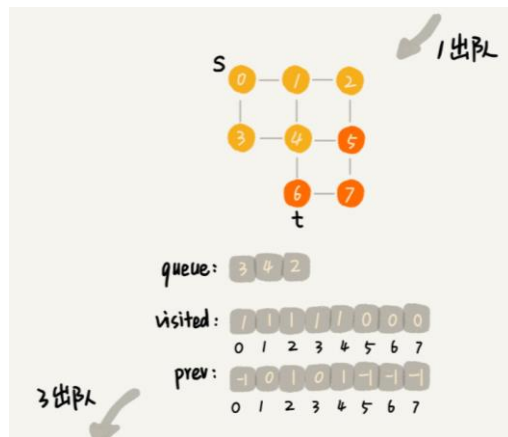
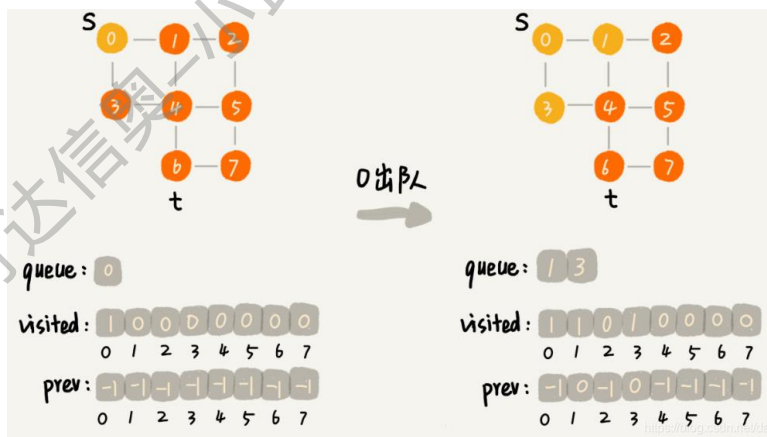
BFS 本质上就是一幅「图」，让你从一个起点，走到终点，问最短路径



广度优先搜索



BFS 本质上就是一幅「图」，让你从一个起点，走到终点，问最短路径



`visited[]` 用来记录已经被访问的顶点，避免顶点被重复访问，如果顶点`q`被访问，那么相应的`visited[q]`就会被设置为`true`；

`queue` 队列用来将遍历的顶点加入队列；

`prev[]` 数组用来记录搜索路径。当从顶点`s`搜索到顶点`t`时，`prev`数组中存储从顶点`s`到顶点`t`所经历的路径。不过是反向存储的。



广度优先搜索



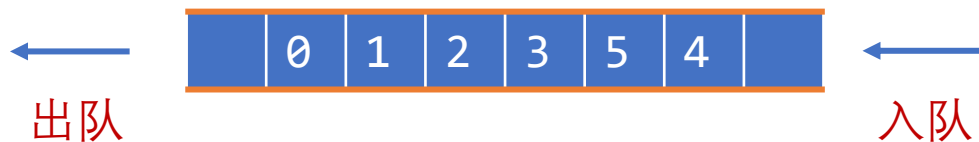
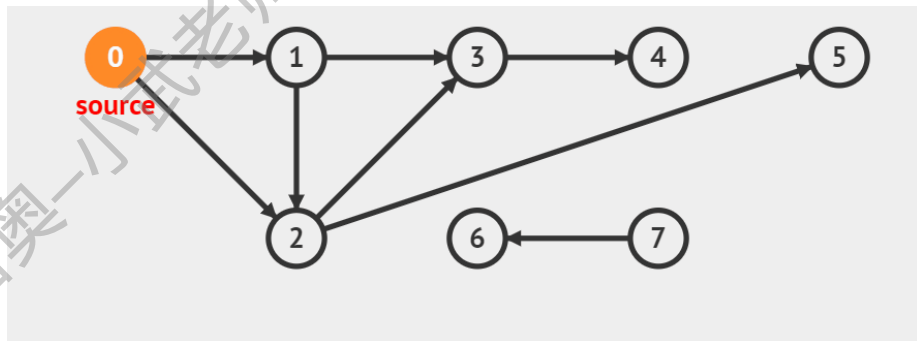
穿个马甲就不认识了？

- 1、走迷宫，有的格子是围墙不能走，从起点到终点的最短距离是多少？
- 2、两个单词，要求你通过某些替换，把其中一个变成另一个，每次只能替换一个字符，最少要替换几次？
- 3、连连看游戏，两个方块消除的条件不仅仅是图案相同，还得保证两个方块之间的最短连线不能多于两个拐点。你玩连连看，点击两个坐标，游戏是如何判断它俩的最短连线有几个拐点的？
- 4、马的遍历，奇怪的电梯等等。

BFS 本质上就是一幅「图」，让你从一个起点，走到终点，问最短路径



广度优先搜索



BFS 的核心数据结构是队列

```
while (队列不为空) {  
    int u = 队首;  
    弹出队首;  
    for (枚举 u 的邻居) {  
        更新数据  
        if (...)   
            添加到队首;  
        else   
            添加到队尾;  
    }  
}
```

BFS 的核心框架

编程实践 Online Judge

P0095. 细胞

P0379 仙岛求药

P0420. 走迷宫

P0094 最少步数





P0095 细胞



4 10

0	2	3	4	5	0	0	0	6	7
1	0	3	4	5	6	0	5	0	0
2	0	4	5	6	0	0	6	7	1
0	0	0	0	0	0	0	0	8	9

```
void dfs(int x,int y){  
    if(x>n||x<1||y>m||y<1||a[x][y]=='0')  
        return;  
    a[x][y]='0';  
    dfs(x+1,y);  
    dfs(x-1,y);  
    dfs(x,y-1);  
    dfs(x,y+1);  
}
```

DFS



P0095 细胞

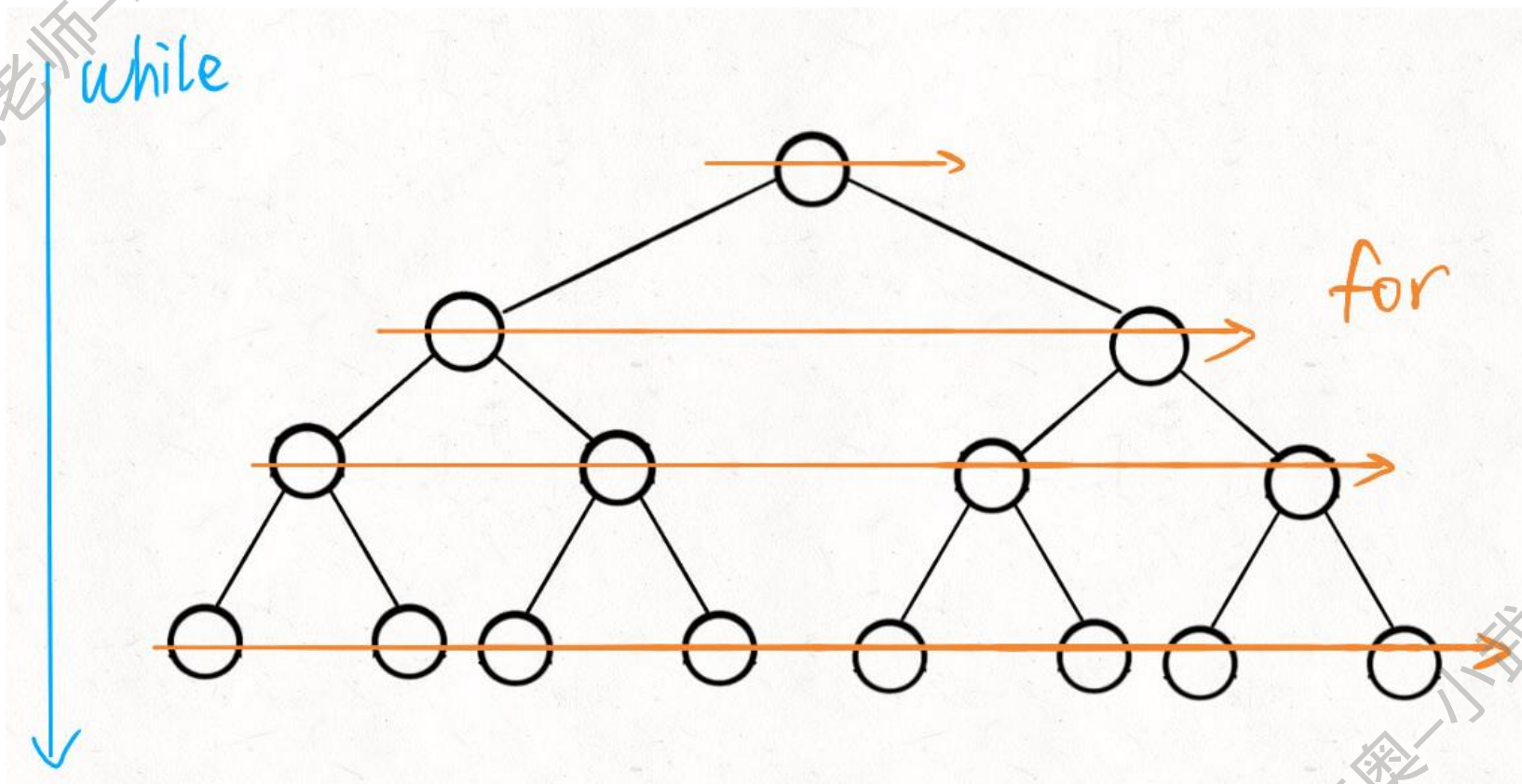


```
void bfs(int x, int y){
    int head = 1, tail = 1;
    q[tail].x = x;
    q[tail].y = y;
    tail++;
    vis[x][y] = 1;
    while(head < tail){
        int x0 = q[head].x;
        int y0 = q[head].y;
        for(int i = 0; i < 4; i++){
            int nx = x0 + d[i][0];
            int ny = y0 + d[i][1];
            if(nx >= 0 && nx < n && ny >= 0 && ny < m && a[nx][ny] != '0' && vis[nx][ny] != 1){ //vis[nx][ny] != 1
                q[tail].x = nx;
                q[tail].y = ny;
                tail++;
                vis[nx][ny] = 1;
            }
        }
        head++;
    }
}
```

广度优先搜索



P0420 走迷宫



广度优先搜索



P0420 走迷宫

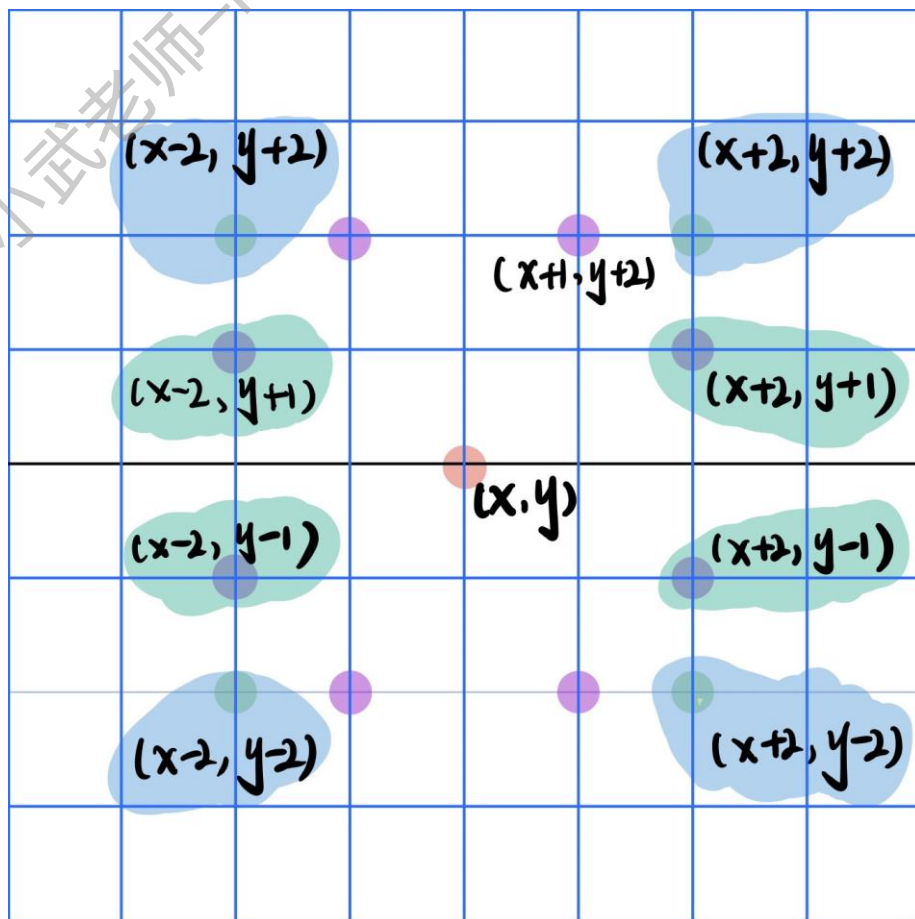


```
void bfs(int sx, int sy, int ex, int ey){
    int head = 1, tail = 1;
    q[tail].x = sx;  q[tail].y = sy;
    q[tail].step = 1;
    tail++;
    vis[sx][sy] = 1;
    while(head < tail){
        int x0 = q[head].x;
        int y0 = q[head].y;
        int step = q[head].step;
        if(x0==ex && y0==ey){
            cout << step; return;
        }
        for(int i = 0; i < 4; i++){
            int nx = x0 + d[i][0];
            int ny = y0 + d[i][1];
            if(nx >= 0 && nx < n && ny >= 0 && ny < m && a[nx][ny] != '#' && vis[nx][ny] != 1){ //vis[nx][ny] != 1
                q[tail].x = nx; q[tail].y = ny;
                q[tail].step = step+1;  tail++;
                vis[nx][ny] = 1;
            }
        }
        head++;
    }
}
```

广度优先搜索



P094 最小步数



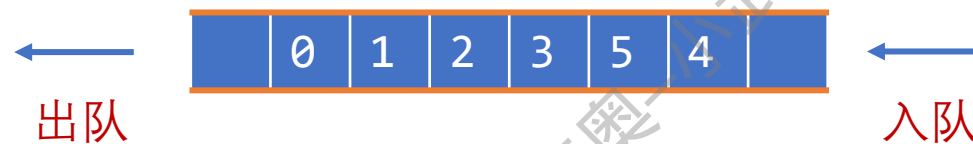
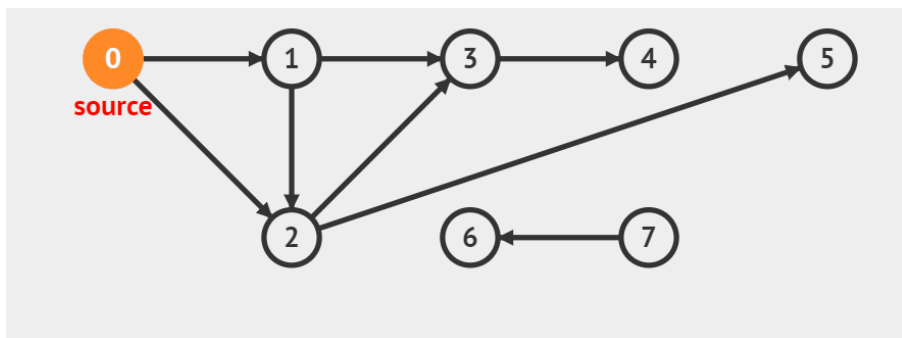
广度优先搜索



BFS总结



- 1、对于数据结构和算法的学习，难的不是掌握原理，而是能灵活地将各种场景和问题抽象成对应的数据结构和算法。比如迷宫可以抽象成图，走迷宫可以抽象成搜索算法。那么，如何将迷宫抽象成一个图？即如何在计算机中存储一个迷宫？
- 2、广搜原理简单，代码实现复杂。深搜适合递归实现，算法原理不易理解，但代码简洁。
- 3、广搜和深搜的算法时间复杂度都是 $O(V+E)$ ，空间复杂度都是 $O(v)$
- 4、广搜需要用到队列这种数据结构，尽量手写队列。



课后习题与实验

Talk is cheap, show me the code !



声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

下节课见啦！

