以走迷宫找出口为例，首先用到的数据结构，队列。

队列用数组实现，两个变量，head和tail，记录队列的头跟尾

增加数据tail++；删除数据head++；head == tail队列为空

用队列标记哪里已经走过，两个数组，一个x坐标，一个y坐标

以当前这个点拓展，往上下左右的方向走，拓展完毕后，就让当前这个点出队，head++

把可以走的x坐标和y坐标记录在队列里，tail++

遍历队列，访问head指向的点。先判断是不是出口，不是的话再拓展

用一个while实现遍历队列

While（head < tail）

{

...

}

以上是用广度优先算法找迷宫出口

很简单，也有点暴力，遍历当前点可以拓展的点。

脱离这道题看广度优先算法的话，就是把最终解当作是一种目标，找到开始状态

拓展开始状态的每一种可能，往最终解方向去靠，题目的限制就是要求每一种状态的操作也尽量相同，不然的话while循环就会有很多if else，会变得复杂。

优化跟深度优先算法一样，减枝，尽可能地减少队列里面的状态，也就是提前知道这种状态拓展出去的状态不可能达到目标，那么就剪掉。队列越短越好。

广度优先算法我是用队列实现的。