



实验舱
青少年编程
走近科学 走进名校

蛟龙四班

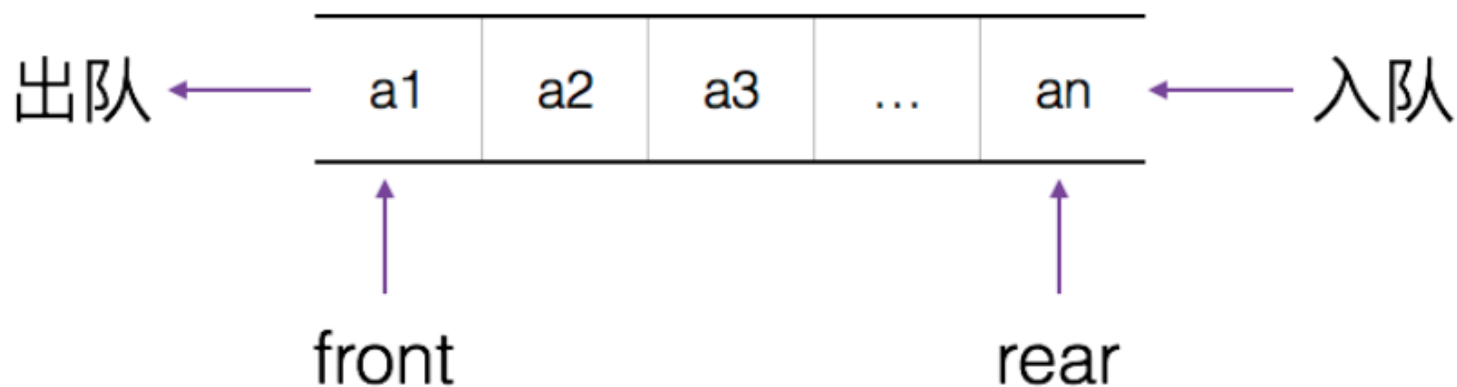
队列与宽度优先搜索

Mas

队列

队列是限定在一端进行插入，另一端进行删除特殊线性表。
就像排队买东西，排在前面的人买完东西后离开队伍（删除），而后来的人总是排在队伍末尾（插入）。

通常把队列的删除和插入分别称为出队和入队。允许出队的一端称为队头，允许入队的一端称为队尾。所有需要进队的数据项，只能从队尾进入，队列中的数据项只能从队头离去。由于总是先入队的元素先出队（先排队的人先买完东西），这种表也称为先进先出（FIFO）表。



队列的常用操作

操作	成员函数
入队	push
出队	pop
访问队首元素	front
大小	size
是否为空	empty

```
1  #include <queue>
2  #include <iostream>
3  using namespace std;
4  int main() {
5      queue<int> q; // 声明一个装 int 类型数据的队列
6      q.push(1); // 入队
7      q.push(2);
8      q.push(3);
9      cout << q.size() << endl; // 输出队列元素个数
10     while (!q.empty()) { // 判断队列是否为空
11         cout << q.front() << endl; // 访问队首元素
12         q.pop(); // 出队
13     }
14     return 0;
15 }
16
```

#417 翻牌

描述

有 n ($n < 100$) 张牌放成一堆，每次从上面取一张牌翻开，再从上面取一张票放到下面。直到取完。

输入

第一行一个整数 n ，接下来 n 个整数，牌面的数字。

输出

按照翻牌的先后次序排列的 n 个整数，两个数之间用空格隔开。

输入样例

```
4
1 2 3 4
```

输出样例

```
1 3 2 4
```

#1165 周末舞会

题目描述

假设在周末舞会上，男士们和女士们进入舞厅时，各自排成一队。跳舞开始时，依次从男队和女队的队头上各出一人配成舞伴。规定每个舞曲能有一对跳舞者。若两队初始人数不相同，则较长的那一队中未配对者等待下一轮舞曲。现要求写一个程序，模拟上述舞伴配对问题。

输入

第一行男士人数 m 和女士人数 n ；

第二行舞曲的数目 k 。

输出

共 k 行，每行两个数，表示配对舞伴的序号，男士在前，女士在后。

样例输入

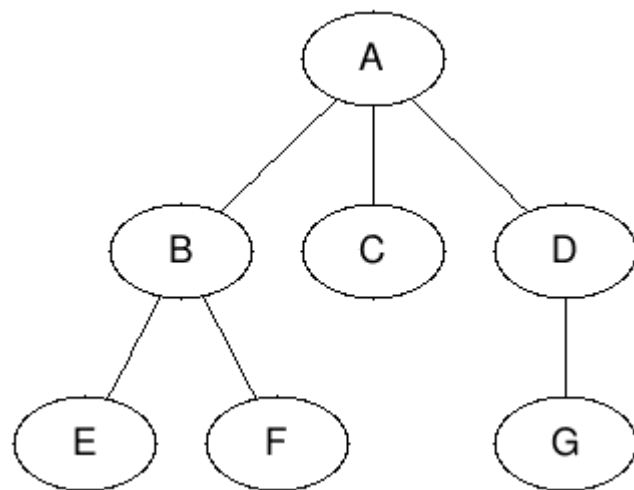
```
4 3
6
```

样例输出

```
1 1
2 2
3 3
4 1
1 2
2 3
```

广度优先搜索

广度优先搜索（Breadth-first-search），又称宽度优先搜索，简称 bfs。与深度优先搜索不同的是，广度优先搜索会先搜索到与起始点距离较近的点，而深搜却是沿着一个分支递归到最后。



宽搜可以理解为，从起点开始一层一层地往外扩展，内层一定会在外层前面被访问到。

与深度优先搜索的对比

深度优先搜索用栈（stack）来实现：

- 把起始节点压入栈中。
- 从栈顶元素，搜索所有在它下一级的元素，把这些元素压入栈中。
- 找到所要找的元素时结束程序

广度优先搜索使用队列（queue）来实现：

- 把起始节点放到队列中。
- 每次从队列的头部取出一个元素，查看这个元素所有的下一级元素，把它们放到队列的末尾。
- 找到所要找的元素时结束程序。

```
void bfs( 起始点 ) {  
    将起始点放入队列中;  
    while (如果队列不为空) {  
        访问队列中队首元素x;  
        删除队首元素;  
        for (x 可以到达的所有点) {  
            if (该点未被访问过且合法) {  
                将该点加入队列末尾;  
            }  
        }  
    }  
    队列为空, 宽搜结束;  
}
```

#1182 走迷宫

【题目描述】

一个迷宫由 R 行 C 列格子组成，有的格子里有障碍物，不能走；有的格子是空地，可以走。

给定一个迷宫，求从左上角走到右下角最少需要走多少步(数据保证一定能走到)。只能在水平方向或垂直方向走，不能斜着走。

【输入】

第一行是两个整数， R 和 C ，代表迷宫的长和宽。（ $1 \leq R, C \leq 40$ ）

接下来是 R 行，每行 C 个字符，代表整个迷宫。

空地格子用 ‘.’ 表示，有障碍物的格子用 ‘#’ 表示。

迷宫左上角和右下角都是 ‘.’。

【输出】

输出从左上角走到右下角至少要经过多少步（即至少要经过多少个空地格子）。计算步数要包括起点和终点。

【输入样例】

```
5 5
..###
#...
#.#.#
#.#.#
#.#..
```

【输出样例】

```
9
```


#1182 走迷宫

思路1:

宽度优先搜索是2一层一层的往外扩展，如果达到终点。那么必然是最短路径。

0	1	1	1	0	0	0	0	0	1
1	0	1	1	0	1	0	1	0	1
1	1	1	0	1	0	0	0	0	0
1	0	0	0	1	0	1	0	1	1
0	0	0	0	0	1	1	1	0	0
1	1	0	1	0	1	0	0	1	0
0	0	1	0	1	1	1	1	0	1
1	1	1	1	0	1	1	0	0	0
0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	0	1	1	0	1

#1168 连通块

【题目描述】

一个 $n \times m$ 的方格图，一些格子被涂成了黑色，在方格图中被标为 1 ，白色格子标为 0 。问有多少个四连通（上下左右四个方向）的黑色格子连通块。四连通的黑色格子连通块指的是一片由黑色格子组成的区域，其中的每个黑色格子能通过四连通的走法（上下左右），只走黑色格子，到达该联通块中的其它黑色格子。

【输入】

第一行两个整数 n, m ($1 \leq n, m \leq 100$)，表示一个 $n \times m$ 的方格图。

接下来 n 行，每行 m 个整数，分别为 0 或 1 ，表示这个格子是黑色还是白色。

【输出】

一行一个整数 ans ，表示图中有 ans 个黑色格子连通块。

【输入样例】

```
3 3
1 1 1
0 1 0
1 0 1
```

【输出样例】

```
3
```

#1168 连通块

思路1:

对每一个不为1且未被标记的格子开始dfs标记,
dfs的次数就是连通块的数目

思路2:

对每一个不为1且未被标记的格子开始bfs标记,
bfs的次数就是连通块的数目



实验舱
青少年编程
走近科学 走进名校

谢谢观看