

数据结构与算法平时考试 1

陈宇琪

2020 年 4 月 19 日

目录

1	层次遍历	2
1.1	题面	2
1.2	输入约定	2
1.3	输出约定	3
1.4	样例 1	3
1.5	样例 2	3
1.6	数据规模约定	3
2	医疗排队	4
2.1	题面	4
2.2	输入约定	4
2.3	输出约定	4
2.4	样例	4
2.5	数据约定	5
2.6	样例说明	5
3	数独问题	6
3.1	题面	6
3.2	输入约定	6
3.3	输出约定	6
3.4	样例 1	7
3.5	样例 2	7
3.6	数据约定	8

1 层次遍历

1.1 题面

给定一棵数，实现层次遍历，并输出层次遍历的顺序，这里假设树的根节点为 1。
为了简化问题，我们采用这种方式来定义树：

- 假设这是一棵二叉树，即一个节点最多只有两个子节点
- 每个节点有一个编号 i ，满足 $1 \leq i \leq n$ 。
- 每个节点有左儿子和右儿子，如果没有的话用 -1 表示。

图片展示的是层次遍历的过程：

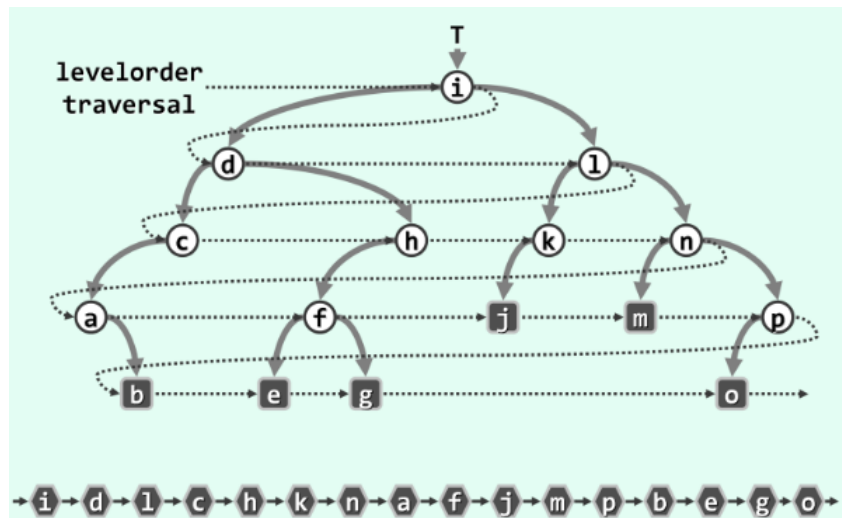


图 1: Level Order

下图展示的是一棵二叉树，**样例 1** 中将给出这棵树的表示方式：

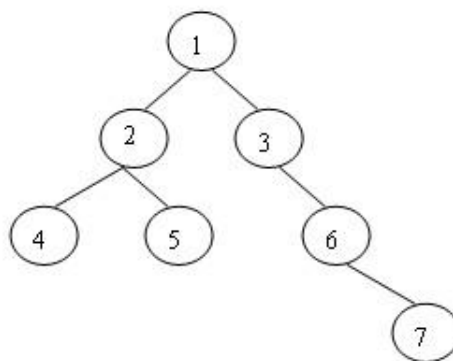


图 2: Tree Sample

1.2 输入约定

输入第一行为一个数 n ，表示节点个数。

接下来 n 行，第 i 行的两个数，表示节点 i 的左儿子和右儿子。

1.3 输出约定

输出 n 个数，表示层次遍历的顺序。

1.4 样例 1

1.4.1 样例输入 1

```
7
2 3
4 5
-1 6
-1 -1
-1 -1
-1 7
-1 -1
```

1.4.2 样例输出 1

```
1 2 3 4 5 6 7
```

1.5 样例 2

1.5.1 样例输入 2

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

1.5.2 样例输出 2

```
1 4 2 3
```

1.6 数据规模约定

对于 30% 数据满足： $1 \leq n \leq 10$ 。

对于 70% 数据满足： $1 \leq n \leq 10^4$ 。

对于 100% 数据满足： $1 \leq n \leq 10^6$ 。

2 医疗排队

2.1 题面

假设一个国家的每个公民都将被分配一个唯一的数字，从 1 到 P (这里的 P 是当前的人口)。它们将被放入一个队列，1 在 2 前面，2 在 3 前面，以此类推。医院将从这个队列中逐一处理病人。一旦市民被服务，他们会立即从队伍的前面移到后面。

当然，有时会出现紧急情况——如果你刚刚被一个蒸汽压路机碾过，你不能等到半个国家去做常规检查后再接受治疗！因此，对于这些（希望是罕见的）情况，可以使用一个快速命令将一个人移到队列的前面。其他国家的相对秩序将保持不变。

给定处理命令和快速命令的顺序，输出当前服务的公民的编号。



图 3: Medical System

2.2 输入约定

第一行两个数 n, p ，表示命令个数和公民个数。

接下来 n 行，每行为一个命令：

- **1**: 表示当前队列首部的公民被服务，被服务后自动放到队列末位。
- **2 x**: 表示将编号为 x 的公民移动到队列首部（不进行服务操作）。
- **3**: 询问当前队列首部的公民的编号（即下一个要被服务的公民编号）。

满足 $1 \leq x \leq p$ 。

2.3 输出约定

对于每个 **3** 命令，输出一个数表示当前队列首部的公民的编号。

2.4 样例

2.4.1 样例输入

```
10 5
2 1
3
2 5
3
```

1
1
1
3
1
3

2.4.2 样例输出

1
5
3
4

2.5 数据约定

对于 30% 数据满足： $1 \leq n, p \leq 10$ 。

对于 70% 数据满足： $1 \leq n, p \leq 5 \times 10^3$ 。

对于 100% 数据满足： $1 \leq n, p \leq 10^6$ 。

2.6 样例说明

开始队列为：1 2 3 4 5

第一个命令执行后队列为：1 2 3 4 5（1 本来就在队首）

第二个命令执行后队列为：1 2 3 4 5，输出 1

第三个命令执行后队列为：5 1 2 3 4

第四个命令执行后队列为：5 1 2 3 4，输出 5

第五个命令执行后队列为：1 2 3 4 5

第六个命令执行后队列为：2 3 4 5 1

第七个命令执行后队列为：3 4 5 1 2

第八个命令执行后队列为：3 4 5 1 2，输出 3

第九个命令执行后队列为：4 5 1 2 3

第十个命令执行后队列为：4 5 1 2 3，输出 4

3 数独问题

3.1 题面

给定一个不完整的数独，要求将数独中空白的部分用 1 到 9 的数字填充。

一个数独的解法需遵循如下规则：

- 数字 1 到 9 在每一行只能出现一次。
- 数字 1 到 9 在每一列只能出现一次。
- 数字 1 到 9 在每一个以粗实线分隔的 3×3 宫内只能出现一次。

下图为一个简单的数独：

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

图 4: Sudo Sample

下图为上面数独的解。

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

图 5: Sudo Solution

3.2 输入约定

输入 9 行，每行 9 个字符，空格用 . 表示。

3.3 输出约定

输出 9 行，每行 9 个字符，如果有多解，输出字典序最小的解。

其中字典序定义为将数独按照行优先比较，即假设两个数独 a 和数独 b ，如果前 i 行都一样，则按照第 $i+1$ 行的字典序大小决定两个数独的大小。

3.4 样例 1

3.4.1 输入样例 1

53..7....
6..195...
.98....6.
8...6...3
4..8.3..1
7...2...6
.6....28.
...419..5
....8..79

3.4.2 输出样例 1

534678912
672195348
198342567
859761423
426853791
713924856
961537284
287419635
345286179

3.5 样例 2

3.5.1 输入样例 2

534678...
6..19534.
1983..567
85.761.23
426853..1
7..924856
961..7284
287419635
..5286179

3.5.2 输出样例 2

534678912
672195348
198342567
859761423
426853791
713924856

961537284
287419635
345286179

3.6 数据约定

对于 30% 数据满足：数独中最多 7 个空格。

对于 70% 数据满足：数独保证有唯一解。

对于 100% 数据满足：数独不保证有唯一解。