

OJ2: 军训排队

问题描述

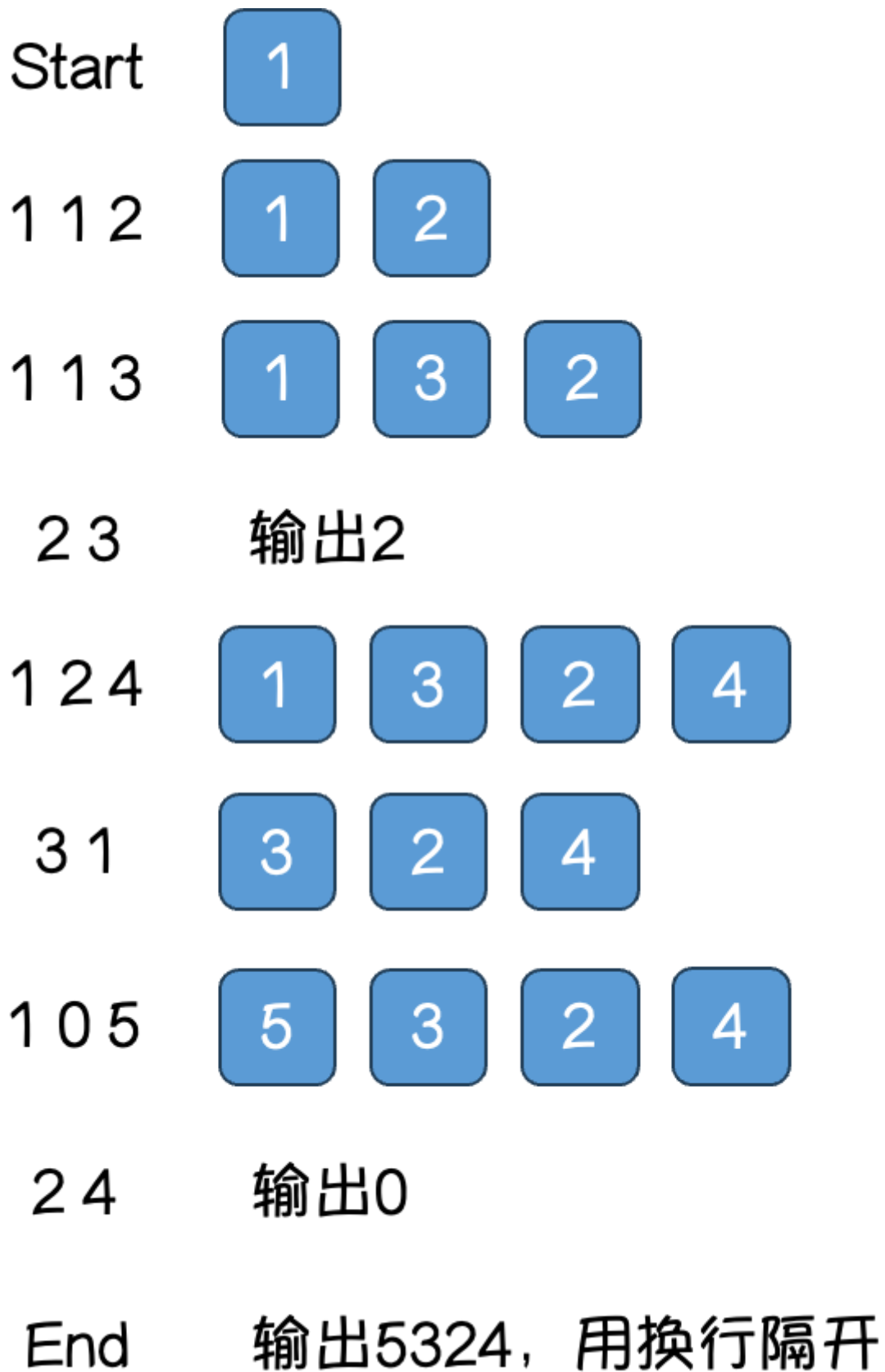
军训时需要将所有同学(包括教官)排成一队，一开始只有教官(编号为1)排在第一个，之后教官会发出如下三种命令：

1 x y 编号为y的同学入队，排在编号为x的同学的后面，若要排在队首，则用0表示x

2 x 编号为x的同学报告其后面的同学编号，若要报告队首同学的编号，则用0表示x，若x在队尾，则报告0

3 x 编号为x的同学出队

要求对于每次命令2，都输出报告同学后面同学的编号，并最后从头到尾输出队列中同学的编号。



输入格式

第一行为一个正整数N，表示总共有几条命令

接下来N行中，每行表示一个命令；每行命令中，数字之间用空格隔开

输出格式

对于每次命令2，都输出一个编号，用换行隔开

最后从头到尾输出队列，用换行隔开

输入样例

```
1 // 例如上图中的命令对应以下输入
2 7
3 1 1 2
4 1 1 3
5 2 3
6 1 2 4
7 3 1
8 1 0 5
9 2 4
```

输出样例

```
1 // 例如上图中的命令会产生以下输出
2 2
3 0
4 5
5 3
6 2
7 4
```

提示

1. 同学编号为小于100000的正整数
2. 命令的条数N为小于200000的正整数
3. 队列中没有人时不会出现命令2和命令3

思路

使用前向后向的双向数组，`queue_next[i]` 和 `queue_prev[i]`，分别记录同学 `i` 的下一个人和前一个人，同时根据题目条件恰好可以定义队列的第一个人为0，最后一个人也为0，完美适配数组操作。当 `i` 出队时，将 `queue_next[i]` 和 `queue_prev[i]` 都标记为-1，表示已经不在队列中。

由此，对于队列的查找、插入和删除在逻辑上和双向链表类似，但是又简单很多，对于某个人 `i`，可以直接通过数组下标定位到他的位置，以及他前后相邻的人，且操作都是常数 $O(1)$ 的，最终的时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$ 。

代码 (C) :

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int N;
7     int cmd1, cmd2, cmd3;
8     scanf("%d", &N);
```

```

9      int queue_next[100001] = {1, 0};
10     int queue_prev[100001] = {1, 0};
11     for (int i = 2; i < 100001; i++)
12     {
13         queue_next[i] = queue_prev[i] = -1;
14     }
15     for (int i = 0; i < N; i++)
16     {
17         scanf("%d", &cmd1);
18         switch (cmd1)
19         {
20             case 1:
21                 scanf("%d%d", &cmd2, &cmd3);
22                 queue_next[cmd3]=queue_next[cmd2];
23                 queue_prev[queue_next[cmd2]]=cmd3;
24                 queue_next[cmd2]=cmd3;
25                 queue_prev[cmd3]=cmd2;
26                 break;
27             case 2:
28                 scanf("%d", &cmd2);
29                 printf("%d\n",queue_next[cmd2]);
30                 break;
31             case 3:
32                 scanf("%d", &cmd2);
33                 queue_prev[queue_next[cmd2]]=queue_prev[cmd2];
34                 queue_next[queue_prev[cmd2]]=queue_next[cmd2];
35                 queue_next[cmd2]=queue_prev[cmd2]=-1;
36
37                 break;
38         }
39     }
40     int i=0;
41     while(queue_next[i]){
42         printf("%d\n",queue_next[i]);
43         i=queue_next[i];
44     }
45     return 0;
46 }

```

最后附上作者的通过图片~

提交详情 (军训排队)

提交者: 2322022010597 创建时间: 2023-10-12 13:25:53

运行结果			分数
			100.00
#	状态	时间	内存
1	Accepted	0 ms	1536 KB
2	Accepted	0 ms	1536 KB
3	Accepted	0 ms	1540 KB
4	Accepted	0 ms	1540 KB
5	Accepted	0 ms	1536 KB
6	Accepted	4 ms	1540 KB
7	Accepted	4 ms	1536 KB
8	Accepted	28 ms	1536 KB
9	Accepted	40 ms	1540 KB
10	Accepted	60 ms	1536 KB