

一道让我有所收获的题（线性表的组织）

洛谷链接 <https://www.luogu.com.cn/problem/P1160>

考察点：线性表的组织、插入删除与查找

题目描述

一个学校里老师要将班上 N 个同学排成一列，同学被编号为 $1 \sim N$ 。他采取如下的方法：

1. 先将1号同学安排进队列，这时队列中只有他一个人；
2. $2 \sim N$ 号同学依次入列，编号为 i 的同学入列方式为：老师指定编号为 i 的同学站在编号为 $1 \sim (i-1)$ 中某位同学（即之前已经入列的同学）的左边或右边；
3. 从队列中去掉 M ($M < N$) 个同学，其他同学位置顺序不变。

在所有同学按照上述方法队列排列完毕后，老师想知道从左到右所有同学的编号。

输入格式

第1行为一个正整数 N ，表示了有 N 个同学。

第 $2 \sim N$ 行，第 i 行包含两个整数 k, p ，其中 k 为小于 i 的正整数， p 为0或者1。若 p 为0，则表示将 i 号同学插入到 k 号同学的左边， p 为1则表示插入到右边。

第 $N+1$ 行为一个正整数 M ，表示去掉的同学数目。

接下来 M 行，每行一个正整数 x ，表示将 x 号同学从队列中移去，如果 x 号同学已经不在队列中则忽略这一条指令。

输出格式

1行，包含最多 N 个空格隔开的正整数，表示了队列从左到右所有同学的编号，行末换行且无空格。

输入输出样例

输入

```
4
1 0
2 1
1 0
2
3
3
```

输出

2 4 1

说明/提示

样例解释：

将同学2插入至同学1左边，此时队列为：21

将同学3插入至同学2右边，此时队列为：231

将同学4插入至同学1左边，此时队列为：2341

将同学3从队列中移出，此时队列为：241

同学3已经不在队列中，忽略最后一条指令

最终队列：241

数据范围

对于20%的数据，有 $N \leq 10$ ；对于40%的数据，有 $N \leq 1000$ ；对于100%的数据，有 $N, M \leq 100000$

解题说明

本题考查点为线性表。由于我们需要对题目进行按照要求采用双向链表进行操作即可。但是如果单纯的用链表会产生TLE等情况。这时想到数组具有查找快的特点，链表具有插入删除快的特点。同时注意到学生的顺序是依次连续增加的。所以可以将二者的优点结合起来，用结构体数组存链表。这样就解决了TLE的问题。

参考代码

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int flag;
    int id;
    struct Node *left;
    struct Node *right;
}queue[100010];

int main() {
    struct Node *head;
    head = &queue[0];
```

```
head->id = 1;
head->flag = 0;
head->left = NULL;
head->right = NULL;
int i;
int n;
scanf("%d", &n);

for (i = 2; i <= n; i++) {
    int k, p;
    struct Node *next = NULL;
    next = &queue[i - 1];
    scanf("%d%d", &k, &p);
    struct Node *temp = &queue[k - 1];
    if (p == 0) {
        next->id = i;
        next->flag = 0;
        next->right = temp;
        if (temp->left != NULL) {
            next->left = temp->left;
            temp->left->right = next;
        }
        else {
            head = next;
        }
        temp->left = next;
    }
    else {
        next->id = i;
        next->flag = 0;
        next->left = temp;
        if (temp->right != NULL) {
            next->right = temp->right;
            temp->right->left = next;
        }
        temp->right = next;
    }
}

int m;
scanf("%d", &m);
for (i = 0; i < m; i++) {
    int temp;
    scanf("%d", &temp);
    queue[temp - 1].flag = 1;
}

struct Node *temp = head;
while (temp != NULL) {
    if (temp->flag != 1) {
        printf("%d ", temp->id);
    }
}
```

```
    }  
    temp = temp->right;  
}  
  
return 0;  
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:%E7%BA%BF%E6%80%A7%E8%A1%A8%E7%9A%84%E5%AE%9E%E7%8E%B0> 

Last update: 2020/05/09 12:20