

## P1160 队列安排

### 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- $N$ 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。

### 输入样例

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

### 输出样例

```
2 4 1
```

## 解析

本题存在两个难点：

- 1、如何快速的定位指定的数据
- 2、如何判断某个数据是否已经被删除

对于问题1，我们可以创建一个指针的数组，以数据本身为索引，迭代器为值，这样我们就能快速定位数据。

对于问题2，我们也可以建立一个数组来维护它的删除状态。

## 编码

```
#include<bits/stdc++.h>

using namespace std;

list<int> nums;

const int maxN = 1e5 + 10;
//存储指针的数组，便于快速定位数据
list<int>::iterator iters[maxN];
//标记当前位置的数据是否被移除
bool erased[maxN];

int main() {
    //将1号同学插入
    nums.push_back(1);
    //记录第一个数据的指针
    iters[1] = nums.begin();
    //n个小朋友
    int n;
    cin >> n;
    //将同学们加入到链表中
    for (int i = 2; i <= n; ++i) {
        int base, opt;
        cin >> base >> opt;
        //找到base的迭代器
        list<int>::iterator it = iters[base];

        switch (opt) {
            //插到右边
            case 1:
                iters[i] = nums.insert(++it, i);
                break;
            //插到左边
            case 0:
```

```

        iters[i] = nums.insert(it, i);
        break;
    }
}
//待删除的数量
int m;
cin >> m;
for (int i = 0; i < m; ++i) {
    int target;
    cin >> target;
    list<int>::iterator it = iters[target];
    //如果当前数据未被删除，则从链表中移除
    if (!erased[target]) {
        nums.erase(it);
    }
    erased[target] = true;
}
//循环打印
list<int>::iterator it;
for (it = nums.begin(); it != nums.end(); it++) {
    cout << *it << " ";
}
return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

