# NOI模拟赛3

saffah

2015 年 5 月 28 日

竞赛时间：2333年3月33日3:33～33:33

| 中文题目名称 | 化学题 | $k$小路径 | 可持久化打字机 |
|---|---|---|---|
| 英文题目名称 | chem | kth | typewriter |
| 输入文件名 | chem.in | kth.in | typewriter.in |
| 输出文件名 | chem.out | kth.out | typewriter.out |
| 每个测试点时限 | 0.5秒 | 1秒 | 1秒 |
| 测试点数目 | 10 | 20 | 20 |
| 每个测试点分值 | 10 | 5 | 5 |
| 内存限制 | 128MB | 128MB | 128MB |
| 是否有部分分 | 否 | 否 | 否 |
| 题目类型 | 传统 | 传统 | 传统 |

阿斯大法好

# 1 化学题

## 1.1 题目描述

小P和小Z是一对非常好的朋友，今天他们在做一道化学题。

这道题目也非常简单，就是求一个链烷烃的一氯取代物的种数。

众所周知，一个链烷烃可以看作一个由H和C构成的树，其中所有H的度数为1，C的度数为4。

对于一个链烷烃，我们可以从它的任何一个H出发进行深度优先搜索，得到的节点序列称为DFS序列。例如甲烷的DFS序列是HCHHH，乙烷是HCHHCHHH（注意，也有其他表示方法例如HCCHHHHH等）。已知了DFS序列，那么链烷烃的结构就确定了。

我们就输入一个链烷烃的DFS序列，求它的不同的一氯取代物的种数。注意我们这里只考虑官能团位置异构，不考虑价键异构、立体异构等。

## 1.2 输入格式

输入文件为chem.in。

只有一行，表示这个链烷烃的DFS序列。

## 1.3 输出格式

输出文件为chem.out。

只有一个整数，表示它的不同的一氯取代物的种数。

## 1.4 样例输入1

HCHHH

## 1.5 样例输出1

1

## 1.6 样例输入2

HCCHHHCHHHCHHH

### 1.7 样例输出2

2

### 1.8 样例输入3

HCCCHHHCHHHCHHHHH

### 1.9 样例输出3

1

### 1.10 样例解释

三个样例分别是甲烷、异丁烷、新戊烷。

### 1.11 数据规模与约定

对于所有数据，输入文件是随机生成的。令$n$为输入串的长度，则：

| 测试点编号 | $n$ |
|---|---|
| 1 | 14 |
| 2 | 17 |
| 3 | 17 |
| 4 | 302 |
| 5 | 302 |
| 6 | 3002 |
| 7 | 3002 |
| 8 | 300002 |
| 9 | 300002 |
| 10 | 300002 |

## 2 $k$小路径

### 2.1 题目描述

给出一个$n$个点，$m$条边的有向图。编号为$i$的边从$s_i$连向$t_i$。保证没有重边但是可能有自环。

编号为$i$的点上有点权$a_i$，保证所有的点权是不超过$p$的正整数。

现在给出$l$，求长度为$l$（经过$l$个点）的路径中，点权序列的字典序前$k$小的所有路径。同一条路径可以多次经过某一个点。

## 2.2 输入格式

输入文件为kth.in。

第一行五个正整数$n, m, p, l, k$。

接下来一行$n$个正整数表示所有的点权$a_i$。

接下来$m$行，每行两个正整数$s_i, t_i$表示一条边。

## 2.3 输出格式

输出文件为kth.out。

依次输出点权序列字典序前$k$小的长度为$l$的路径，每行一个。输出路径的方法是依次输出路径上每个点的点权。

如果长度为$l$的路径不足$k$条，则依次输出所有路径，再输出一行END。

## 2.4 样例输入1

```
3 4 2 3 100
1 2 2
1 2
1 3
2 3
3 2
```

## 2.5 样例输出1

```
1 2 2
1 2 2
2 2 2
2 2 2
END
```

## 2.6 样例输入2

```
2 1 2 1 1
1 2
```

1 2

### 2.7 样例输出2

1

### 2.8 样例解释

注意要输出的是点权而不是编号。比较字典序也是点权而不是编号。

### 2.9 数据规模与约定

存在30%的数据，$n \leq 5, l \leq 10$。
存在30%的数据，$k \leq 10000$，且满足任意两点的点权都不同。
存在40%的数据，$p \leq 100, k \leq 10000$。
以上三类数据两两没有交集，构成了全部数据。
对于100%的数据，$n, p \leq 500, m \leq n^2, l \leq 100, k \leq 100000$。

## 3 可持久化打字机

### 3.1 题目描述

小P和小Z有一种神奇的打字机，说它神奇是因为它支持可持久化。这种打字机是这样工作的：

打字机时刻维护着一个字符串，初始为空串。一共有$q$次操作，第$i$次操作可以是：

- 1 $c$，其中$c$是一个小写字母。接到这个命令后，打字机会在当前字符串后添加这个字符$c$。
- 2 $x$，其中$x$是小于$i$的自然数（可以为0）。接到这个命令后，打字机会把当前字符串变成进行$x$次操作后那时的字符串。

例如，考虑这个操作序列：$q = 3$，操作为：1 $a$, 2 0, 1 $b$，那么产生的3个字符串就分别为a,空串,b。

现在我们有两台这样的打字机，称为$A$和$B$。我们用$A_i$表示打字机$A$产生的第$i$个字符串，$B_j$表示打字机$B$产生的第$j$个字符串。如果$A_i$在$B_j$中出现了$k$次，那么我们获得$i \times j \times k$的得分。

求总得分对1000000007取模的结果。

注意：对于两个字符串$s,t$，如果$s$是空串，则我们认为$s$在$t$中出现的次数为$t$的长度+1。

## 3.2 输入格式

输入文件为typewriter.in。

第一行有一个整数$q$，表示打字机$A$的操作次数。

接下来$q$行，每行格式为1 $c$或2 $x$，表示第一个打字机的每一次操作。

接下来用相同的方式描述了打字机$B$的操作。

## 3.3 输出格式

输出文件为typewriter.out。

输出一个整数，即总得分对1000000007取模的结果。

## 3.4 样例输入

```
3
1 a
2 0
1 b
2
1 a
1 b
```

## 3.5 样例输出

```
24
```

## 3.6 样例解释

$1\times 1\times 1+1\times 2\times 1+2\times 1\times 2+2\times 2\times 3+3\times 1\times 0+3\times 2\times 1=24$。

## 3.7 数据规模与约定

令$n$为两个打字机的操作序列长度的较大值：

对于20%的数据，$n\leq 100$。

对于40%的数据，$n \leq 500$。
对于70%的数据，$n \leq 3000$。
对于100%的数据，$n \leq 300000$。
均匀分布着一半的数据满足所有出现的小写字母均为a。
所有数据都是用某种方法随机生成的。