浅析字母树
在信息学竞赛中的应用

浙江省绍兴市第一中学

董华星

二〇〇八年十二月

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

目 录

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

摘要

字母树（又叫单词查找树、Trie 树，Trie Tree），构造简单，并能很好地处理和“串”相关的检索（Retrieval）问题。本文从此着手，归纳了笔者的学习经验，详细论述了其在快速查找、排序、优化转移等方面的常见应用，并对其扩展做了提示。

关键字

字母树 快速检索 “串”排序 无效转移 最长公共前缀问题

前言

当我们走进图书馆的阅览室寻找书时，会不由自主地根据书架上的分类标签寻找自己所喜好的书籍；当打开电脑中的资源管理器时，我们会看到一层一层的目录结构。它们的存在，方便了我们生活中的一个重要的问题——检索。

图一 资源管理器

在信息学竞赛（Olympiad in Informatics，简称 OI）的学习过程中，我们也经常会遇到关于“检索”的问题。而通常采用的不借助任何数据结构（Data Structure）的的枚举方法，虽然简单易写，但往往存在着效率低下的弊端。那我们如何才能通过简单的途径提高算法中的检索效率？

接下来笔者将浅析并推荐一种数据结构——字母树，借助它我们便能够很好地处理与“串”相关的检索问题。

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

正文

1. 预备知识

为了更深入地让读者了解这种数据结构，本文首先将对其进行一番具体介绍并说明其构造。

1.1. 试题分析

例一 单词查找树[1]

我们需要将一个确定的单词列表建出一棵单词查找树，满足

1. 根结点不包含字母，除根结点外的每个都仅含一个大写英文字母；

2. 从根结点到某一结点，路径上经过的字母依次连起来所构成的字母序列，称为该结点对应的单词。单词列表中的每个词，都是该单词查找树某个结点所对应的单词；

3. 在满足上述条件下，该单词查找树的结点数最少。

统计出该单词查找树的结点数目。

输入格式

若干行单词，描述单词列表。

输出格式

一个数字，表示结点数目。

样例输入

A

AN

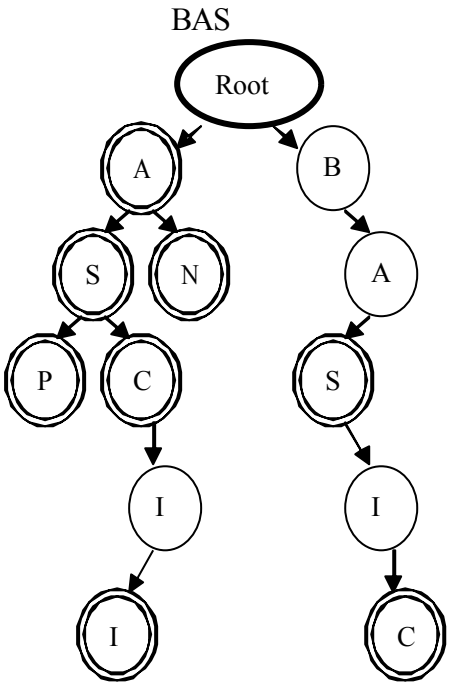
ASP

AS

ASC

ASCII

[1] 第十七届全国青少年信息学奥林匹克竞赛（NOI2000）二试试题



图二 样例示意

BASIC

样例输出

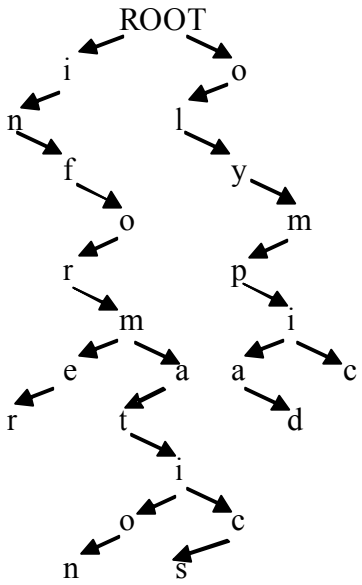
13

解析

题目对这种数据结构的限制内容较多，又要满足“该单词查找树的结点数最少”，故贪心即可。即，对于树中的每个结点，它的包含相同字母的所有儿子都应当合并在一起。按照以上所有条件模拟出这棵树，统计一下其结点个数即可。

1.2. 结构简介

其实，上题所提的这种数据结构便是本文要讲的“字母树”。既然叫做“字母树”，顾名思义，那一定是一种树形结构。而且它还是一棵有根树（Rooted Tree）。其结构特点与上题描述及解析基本一致，其中：



图三 字母树

1. 根节点不包含任何字母，而除它以外的每个结点都仅包含一个字母（事实上也可以将一个数字看成字母来建字母树）；

2. 从根结点到某个结点，路径上经过的字母依次连接起来所构成的字母序列，称为该结点对应的单词。单词列表中的每个词，都是该字母树某个结点所对应的单词；

3. 每个结点的所有儿子包含的字母各不相同。

单词列表为“inform”、“informer”、“information”、“informatics”、“olympic”以及“olympiad”对应的字母树可以是如图三所示。例如，保存“informer”

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

和“information”时，由于它们的前六个字母是相同的，所以希望它们共享这些字母，而只对剩下的部分进行分开存储。可以很明显地发现，字母树很好地利用了串的公共前缀，节约了存储空间[2]。

字母树的插入（Insert）、删除（Delete）和查找（Find）都非常简单，用一个一重循环即可，即第 i 次循环找到前 i 个字母所对应的子树，然后进行相应的操作。实现这棵字母树，我们用最常见的数组保存即可，当然也可以开动态的指针类型。至于结点对儿子的指向，一般有三种方法：

1. 对每个结点开一个字母集大小的数组，对应的下标是儿子所表示的字母，内容则是这个儿子对应在大数组上的位置，即标号；

2. 对每个结点挂一个链表，按一定顺序记录每个儿子是谁；

3. 使用左儿子右兄弟表示法记录这棵树。

三种方法，各有千秋。第一种易实现，但实际的空间要求较大；第二种，较易实现，空间要求相对较小，但比较费时；第三种，空间要求最小，但相对费时且不易写。但总的来说，几种实现方式都是比较简单的，只要在做题时加以合理选择即可，本文不再赘述。

从以上介绍可以明显看出，字母树是一种简单的数据结构。

[2] 这里是指字符串中实际出现的字符的保存量，下同

2. 字母树在串的快速检索中的应用

字母树能很好地利用串的公共前缀，节约了存储空间。既然字母树存储东西的代价少了，那么用它来检索是否应该有着比较高的效率？下面请看例题。

例二 查找生词

给出 N 个单词组成的熟词表，以及一篇全用小写英文书写的文章，请你按最早出现的顺序写出所有不在熟词表中的生词。

输入格式

第一行，一个数字 N;

接着 N 行，每行一个熟词表中的单词；

接着一行，一篇文章。

输出格式

若干行，每行一个生词。

样例输入

5

one

dollar

hundred

thousand

it

it costs one thousand one hundred and one yuan.

样例输出

costs

and

yuan

解析

本题的解法较多。

一种解法是：先将所有熟词用一个数组记录下来。接着枚举文章中的每个单词，对于对应的单词，枚举熟词表，检查它是否已经存在，不存在则输出并且将

它存到熟词表的末尾。反复操作即可。这种算法最简单，也是最容易被人接受的，但时间复杂度明显较大。

另一种解法是：采用哈希（Hash）[3]，对于所有的熟词先建立映射关系。再枚举文章中的每个单词，看看是否已经被哈希了，没有则输出并为其进行哈希。这种做法也比较常见，若哈希函数设计得比较好且合理解决哈希冲突，时间复杂度接近读入复杂度。

既然问题是查找一个单词是否为熟词，正好符合字母树的功能，故又得到了一种解法：对于所有熟词先建出字母树，对于所有对应单词为熟词的结点进行标记。接着枚举所有文章中的单词，去字母树中查找一下对应的结点是否标记，没有标记则输出，顺便把这个单词插入到字母树中并对相应结点进行标记。这种做法很好利用了字母树的特性，我们采用本文之前提到的第一种存储方式，时间复杂度仅为读入复杂度，且常数较小。

比较上述三种解法，可以发现：最直接的做法效率很低；使用哈希，虽然时间复杂度可以接近读入复杂度，但需要借助良好的哈希函数；而使用字母树，由于它的各项操作的时间复杂度均较小，总复杂度可以做到读入复杂度。可以很容易地发现字母树在检索方面的存在着较大优势。

[3] 可参考 IOI2006 国家集训队论文《Hash 函数的设计优化》，作者李羽修
以及 IOI2007 国家集训队论文《Hash 在信息学竞赛中的一类应用》，作者杨弋

3. 字母树在“串”排序方面的应用

既然字母树在检索方面的有着较大优势，很自然便让我们想到拿它来实现字典的索引。翻一翻英语字典，可以发现所有单词表中的单词是按字典序排列的，我们的字母树是否可以实现“串”的排序呢？答案是肯定的。

例三 感谢名单

给你 N 个互不相同的仅由一个单词构成的英文名，让你将它们按字典序从小到大排序输出。

输入格式

第一行，一个数字 N;

接着 N 行，每行一个英文名。

输出格式

共 N 行，每行一个英文名，描述排序后的结果。

样例输入

```
5
Reagan
Bush
Clinton
Bush
Obama
```

样例输出

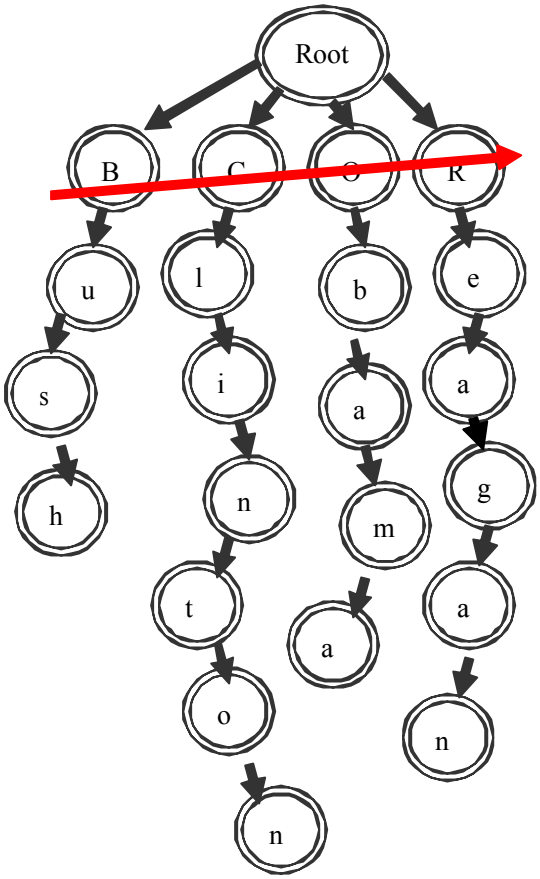
```
Bush
Bush
Clinton
Obama
Reagan
```

解析

对于排序问题，解题者一般首先都会想到那些常用的排序算法——选择排序（Selection Sort）、冒泡排序（Bubble Sort）、归并排序（Merge Sort）、快速排序

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

（Quick Sort）[4]等。如果用归并排序来排一些比较小的数字，因为比较两个数字大小的时间复杂度是 O(1)的，所以总复杂度是 $O(M\log_2 M)$。这里，两个字符串比较大小，直接比的话，最坏复杂度是 O(|S|)的，因此排序的总复杂度变成了 $O(M|S|\log_2 M)$，虽然我们可以用一些方法对其进行优化，但是复杂度还是远远大于读入复杂度。



图四 样例解答

尝试着用字母树进行排序，仍旧采用上文提到的第一种存储方式对所有字符串建树，这棵树的每个结点的所有儿子也就很显然地按照其字母大小排序。接着对这棵树进行先序遍历（又叫前根遍历等，Preorder Traversal），每次遍历到一个对应单词是实际存在的结点就输出这个单词即可。当然，对于字母树的每个结点应该记录一下对应单词出现了几次，避免同一个单词漏输出的情况发生。

用这种做法，所有串被读入与输出各一次，遍历整棵树的代价是树的结点数乘以常数 26，因此，其总复杂度是线性的。

到这里，读者不难惊喜地发现字母树不仅可以做到快速索引，还支持了一种特殊排序算法——“串”排序。而我们知道将数据有序化后能为我们带来更高效的检索。

[4] 可参考《算法艺术与信息学竞赛》，作者刘汝佳、黄亮

4. 字母树在减少无效转移方面的应用

字母树能够提供快速的索引功能，极大地方便了解题者对单词的查找，只要充分利用这一点，它还能够在递推（Recursion）中一显身手。

例四 文章解密

给出有 N 个单词的字典，和一篇长 L 的没有标点符号的文章。问这文章有多少种解释方式。

输入格式

第一行，一个数字 N;

接下来 N 行，每行一个单词，描述字典内容；

接下来若干字符，描述文章。

输出格式

一个数字，表示所求答案并对 12345679 取模。

样例输入

5

ab

cb

bc

ba

a

abcba

样例输出

2

解析

采用递推解决本题。设 $F[i]$为到文章的第 i 个字母结束的解释方式数。每次枚举接下来的一个单词的长度 x，若第 i+1 至 i+x 确为字典中的一个单词，则增加一个递推为 $F[i+x]+=F[i]$。最后答案即为 $F[L]$。那么，此题的复杂度主要在于如何确定一个串为一个单词。

当然，可以采用逐个且逐位比较的判断方式，但是总时间复杂度明显较低，

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

为 O(NZ^2)。

解题者想到“如何确定一个串为一个单词”实际是一个匹配问题。于是，可以考虑用 KMP（Knuth Morris Pratt）算法[5]对每个字典中的单词预先在文章中找到所有的匹配位置。这样，算法总复杂度可以写成 O(NZ)或 O(L^2)，而且常数较大。当然也可以考虑用哈希来处理匹配问题，不过复杂度也并未有所改观。

此时，读者应当想到，可以用字母树来确认一个单词是否存在。先将所有字典中的单词建成字母树，并对字母树上相应位置进行标记。接着，依照一开始的做法，每次先枚举一个位置 i，再枚举接下来的一个单词的长度 x，边枚举单词的长度边在字母树上进行查找，如果当前结点对应的单词是字典中的，则进行相应的转移。此算法的复杂度为 O(LD)，这里的 D 是字母树的深度。当 N 和 L 同级别时，乍一看最坏复杂度和之前所提的利用 KMP 算法或哈希相当。但是，我们可以意识到，当枚举单词长度 x 时，如果在字母树上已经找不到对应结点，便可退出对单词的枚举，即结束了当前 i 下对转移的枚举。这样，在字母树本身常数较小的基础上，还有效减少了许多冗余操作，使程序速度再次得到大大提高。

由上述例题可以看到，字母树提供的快速“检索”，还能够在一定程度上减少一些试题采用递推、动态规划（Dynamic Programming，简称 DP）等算法时的无效转移，加快程序运行速度。

[5] 可参考 Matrix67 博客 2006 年 11 月 29 日志 http://www.matrix67.com/blog/archives/115

5. 字母树在最长公共前缀问题的应用

本文前面重点论述的是字母树在检索方面的功能，但这功能的由来实质是利用了它对串的公共前缀的处理。提到“公共前缀”以及“检索”，读者不难想到一个经典问题，即串的最长公共前缀（Longest Common Prefix，简称 LCP）问题。字母树也能够很好处理该问题。

5.1. 串的最长公共前缀问题

例五 串的最长公共前缀

给出 N 个小写英文字母串，以及 Q 个询问，即询问某两个串的最长公共前缀的长度是多少。

输入格式

第一行，两个数字 N 和 Q;

接下来 N 行，每行一个字母串；

接下来 Q 行，每行一个两个数字，表示对这两个编号的串进行询问

输出格式

Q 行，每行是对应问题的答案。

样例输入

```
2 2
abc
abd
1 1
1 2
```

样例输出

```
3
2
```

解析

本题有多种解法，比如可以用 New LCP[6]、后缀树（Suffix Tree）[7]等。现

[6] 可参考 IOI2006 国家集训队作业《New LCP（原创）》，作者陈启峰
[7] 可参考 IOI2003 国家集训队论文《寻找最大重复子串》，作者林希德

在大致介绍一下字母树的做法。

同样，首先对所有的串建立其对应的字母树。此时发现，对于两个串的最长公共前缀的长度即它们所在结点的公共祖先个数，于是，问题就转化为了离线（Offline）的最近公共祖先（Least Common Ancestor，简称 LCA）问题。

而最近公共祖先问题同样是一个经典问题，可以用下面几种方法：

1. 利用并查集（Disjoint Set）[8]采用经典的 Tarjan 算法；

2. 求出字母树的欧拉序列（Euler Sequence）后，就可以转为经典的最小值查询（Range Minimum Query，简称 RMQ）[9]问题了；

3. 对于字母树上的每个结点，递推求出其所有向上 2^k 后的祖先。查找两个点的最近公共祖先就可以通过它们同时快速地向上跳跃尽可能大的距离得到了。

……

5.2. 数字相关的公共前缀问题

通过上述的介绍，可以发现字母树可以在最长公共前缀和最近公共祖先之间起到桥梁作用。而对于字符串，研究公共前缀就可以得到上述问题，那对于数字研究公共前缀是怎么一回事呢？下面再来看一个例题。

例六 近似问题

两个实数都保留小数点后相同位数 k 后，两数一致，则 k 称为它们的公共精度。这里，k 不超过两数的小数位数的最小值。其中，最大的 k 叫做它们的最长公共精度。现在，有 N 个不足 1 的非负实数，让你求出一个数字 x，使得这个数和所有数字的最长公共精度之和最大。

输入格式

第一行，一个整数 N;

接着 N 行，每行一个小数。

[8] 可参考《算法艺术与信息学竞赛》，作者刘汝佳、黄亮

[9] 可参考 IOI2007 国家集训队论文《RMQ 与 LCA 问题》，作者郭华阳

输出格式

任意一个满足要求的 x。

样例输入

4

0.315

0.31

0.314

0.3

样例输出

0.315

解析

本题中的保留小数是直接保留，并不需要四舍五入，这就为我们的解题带来了方便。将每个数字的小数部分作为一个字符串，用这些串建起一棵字母树。一个很显然的结论是，最后的答案一定是这棵树上某个结点对应的单词。在建树的时候顺便统计出以每个结点对应单词为前缀的字符串个数 v。而以某个单词为我们所求的 x 的最长公共精度和，就是这个单词经过的每个结点的对应的 v 之和。因此，最后只要遍历整棵树，更新一下答案即可。时间复杂度也仅是读入复杂度。

而对于本题，若采用别的解法，并不能够简单快捷并且高效的得到解决。

上题对小数部分的保留是直接保留。对于保留方式是需要进行四舍五入的[10]，本文不再介绍，但也是可以使用字母树进行求解的。这将作为思考留给读者。

[10] POJ 2007 年 11 月月赛第一题 http://acm.pku.edu.cn/JudgeOnline/problem?id=3464

6. 拓展思考

6.1. 字母树应用的扩展

上一节内容的最后一段中说到“字母树的相关的直接应用”。读者一定会想，既然是“直接”，那一定还有“间接”。不错，字母树的应用还远远没有结束。

在例四的解析中，我们提到“‘如何确定一个串为一个单词’实际是一个匹配问题”。这里，“匹配”就很容易让我们想到 KMP 算法，而字母树也可以通过其过硬的检索功能处理该问题。如果把这两者结合起来，会得到什么？相信不少读者已经知道，那就是功能更加强大的一种新的数据结构——AC 自动机（Aho-Corasick Pattern Matching Machine）[11]。

本文对 AC 自动机不再介绍，读者可以参看 2006 年国家集训队成员王赟（My Accept Is Going On，即 Maigo）的论文。

6.2. 字母树的局限

宋代戴复古的《寄兴》中便出有成语“金无足赤，人无完人”。字母树虽然存在诸多优点，但还是有其一定局限。

由于字母树所解决的题目，串对应的字符集一般较小，因此由字符集大小带来的各常数时间和空间复杂度，我们在实践中一般将其忽略。

因其良好的时间复杂度和空间复杂度是建立在字符集较小的前提下的。虽然，字母树的时间和空间复杂度随着其实现方式可以进行一定的转化，但是，针对处理的字符集较大并且要求的时间复杂度也较优的题目它一般不能适用。

所以，我们解题在选用字母树前，应当对题目中数据范围和时间限制加以一定分析。而这一步也是我们做任何题目所需要的。

[11] 可参考 IOI2006 国家集训队论文《Trie 图的构建、活用与改进（第三版）》，作者王赟

7. 总结

字母树是一种好懂、好想、好写、好用的数据结构。它合理利用串的公共前缀来节约内存，在“检索”方面存在着很大优势。

而字母树的所有应用都围绕着其特点展开，除基本的查找外，还能进行字符串的排序，减少递推、动态规划等算法中的无效转移并达到加速的目的，并且它也在“最长公共前缀”这一经典问题的解决中占有一席之地，而本文未作介绍的基于字母树的 AC 自动机有着更强大的应用。可以发现，字母树是一种简单并且全能的数据结构，而它的这些应用都与它的“检索”功能密切相关。当然，不是所有题目都能使用字母树，这需要我们对题目进行仔细分析并合理选择。

字母树，简单，但堪称经典。当做到和关于“检索”的题时，请读者不妨考虑一下，字母树能否为解题带来便利。也许，读者就能够从中获得更多的喜悦。

参考文献

[1] 刘汝佳、黄亮 《算法艺术与信息学竞赛》 [M] 北京：清华大学出版社

[2] 杨弋 《Hash 在信息学竞赛中的一类应用》 [A] IOI2007 国家集训队论文 [C] 中国计算机学会

[3] Thomas H.Cormen 等 《算法导论》 [M] 北京：机械工业出版社

[4] 《KMP 算法详解》 [EB/OL] http://www.matrix67.com/blog/archives/115，2006-11-29

[5] 郭华阳 《RMQ 与 LCA 问题》 [A] IOI2007 国家集训队论文 [C] 中国计算机学会

[6] 《金无足赤，人无完人》 [EB/OL] 百度百科，2008-08-03

附录

1. 单词查找树

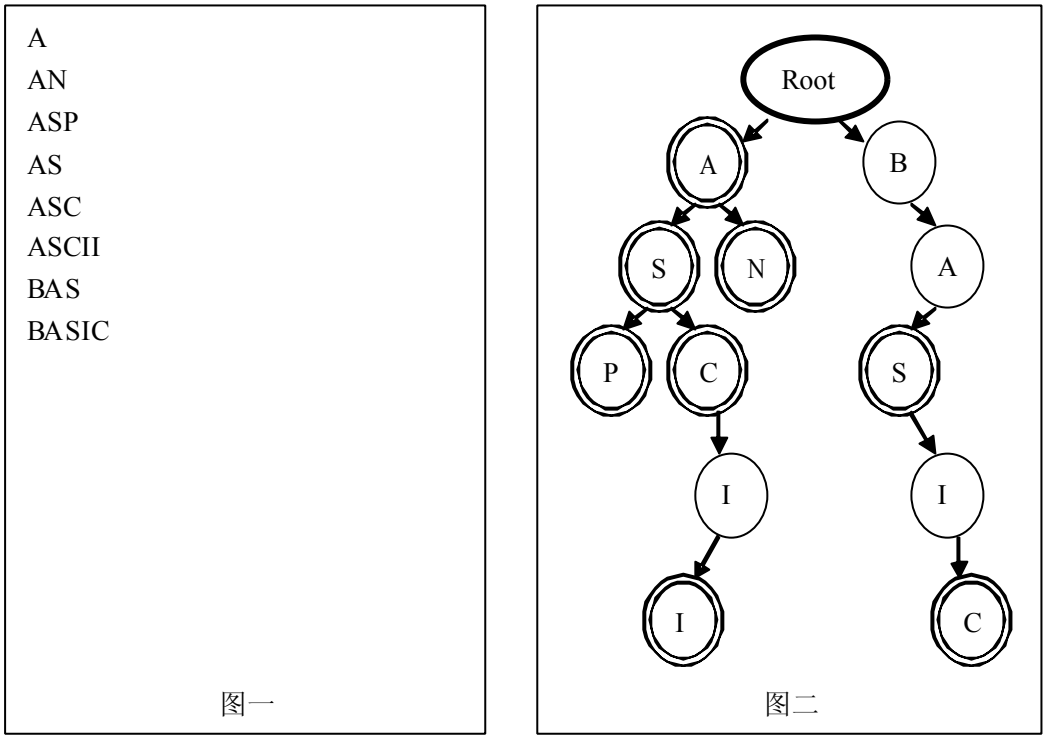
在进行文法分析的时候，通常需要检测一个单词是否在我们的单词列表里。为了提高查找和定位的速度，通常都要画出与单词列表所对应的单词查找树，其特点如下：

根节点不包含字母，除根节点外每一个节点都仅包含一个大写英文字母；

从根节点到某一节点，路径上经过的字母依次连起来所构成的字母序列，称为该节点对应的单词。单词列表中的每个词，都是该单词查找树某个节点所对应的单词；

在满足上述条件下，该单词查找树的节点数最少。

例：图一的单词列表对应图二的单词查找树



对一个确定的单词列表，请统计对应的单词查找树的节点数（包括根节点）

输入格式

该文件为一个单词列表，每一行仅包含一个单词和一个换行/回车符。每个

单词仅由大写的英文字符组成，长度不超过 63 个字符。文件总长度不超过 32K，至少有一行数据。

输出格式

该文件中仅包含一个整数和一个换行/回车符。该整数为单词列表对应的单词查找树的节点数。

样例输入

```
A
AN
ASP
AS
ASC
ASCII
BAS
BASIC
```

样例输出

```
13
```

试题来源

第十七届全国青少年信息学奥林匹克竞赛。

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

2. 感谢名单

好不容易，小 X 大学毕业了。在朋友们的朋友们的齐心帮助下，他在 Y 公司得到了一份相当不错的工作。工作虽然辛苦，但非常符合他的胃口，而且一年下来还能获得一份高薪。

快过年了，小X 便打算在那些天好好请朋友们吃一顿。他想列了一份感谢名单。名单上的所有人，都是他的邀请对象。但是，怕朋友们误会名单上名字的顺序和其“功劳”有关，小 X 决定讲这些名字按其字典序排序。

这下，他就需要你帮忙了。他希望你能将他已有的这份名单中的所有名字按字典序升序排列。

输入格式

第一行，一个数字 *N*，表示名字数目；

接着 *N* 行，每行一个英文名。名字首字母大写，其余小写，中间没有空格。

输出格式

共 *N* 行，每行一个名字，描述排序后的结果。

样例输入	样例输出
5 Reagan Bush Clinton Bush Obama	Bush Bush Clinton Obama Reagan

试题来源

不明

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province

感谢

感谢时刻关心我的亲人和辛勤培育我的老师。

感谢鼓励并帮助我的各位朋友。

感谢给我这次表现机会的中国计算机学会和 IOI2009 中国国家集训队教练。

Huaxing DONG
Shaoxing One Senior High School in Zhejiang Province