



**《程序设计原理》实验报告**

**——问题 A: 本质不同的子串**

**学 号 3020205094**

**姓 名 刘宏伟**

**学 院 智能与计算学部**

**年 级 2021级**

**任课教师 杜朴风**

**2022年 12月 11日**

# 实验目的

选择一个算法，学习并使用它求解字符串环中本质不同的子串数目。（这里我使用的是后缀自动机）

# 二．实验内容

1. **设计思路**

输入n（数据组数），接下来n行，每行一个s（字符串）。

对于每一组数据，用一个字符串存输入的s，长度为m。然后遍历s的0号位到m-1号位，再从0号位遍历到m-2号位（这样组成的字符串相当于一个字符串环），构建后缀自动机，然后利用后缀自动机的性质，累加长度为1到m的所有本质不同子串的数目，得到answer。输出answer，回车。

1. **代码实现**（写上必要的注释）

#include <bits/stdc++.h>

using namespace std;

// 因为每组字符串的长度不超过500，可知后缀自动机状态总数不会超过2\*500-1=999，这里取2000保险

#define MAXSIZE 2000

int main() {

// 定义并输入数据组数

int n;

cin >> n;

while (n--) {

// 定义并输入字符串

string s;

cin >> s;

int s\_length = s.length();

// 定义并初始化要用到的数组

int len[MAXSIZE], lnk[MAXSIZE], cnt[MAXSIZE], nxt[MAXSIZE][26], sub[MAXSIZE], c[MAXSIZE];

memset(c, 0, sizeof c);

memset(len, 0, sizeof len);

memset(lnk, 0, sizeof lnk);

memset(cnt, 0, sizeof cnt);

memset(nxt, 0, sizeof nxt);

// 定义并初始化答案，当前状态指针，last指针

int answer = 0, index = 1, last = 1;

// 遍历拓展字符串以等效字符串环

for (int i = 0; i < 2 \* s\_length - 1; i++) {

// 构造后缀自动机并记录各种状态，ch为当前要插入的字符

int x = ++index, ch = s[i % s\_length] - 'a';

len[x] = len[last] + 1;

sub[x] = 1;

int p;

for (p = last; p && !nxt[p][ch]; p = lnk[p])

nxt[p][ch] = x;

if (!p)

lnk[x] = 1, cnt[1]++;

else {

int q = nxt[p][ch];

if (len[p] + 1 == len[q])

lnk[x] = q, cnt[q]++;

else {

int nq = ++index;

len[nq] = len[p] + 1;

lnk[nq] = lnk[q];

memcpy(nxt[nq], nxt[q], sizeof nxt[q]);

for (; p && nxt[p][ch] == q; p = lnk[p])

nxt[p][ch] = nq;

lnk[q] = lnk[x] = nq;

cnt[nq] += 2;

}

}

last = x;

}

// 更新各个节点状态

for (int i = 2; i <= index; i++) {

c[len[lnk[i]] + 1]++;

c[len[i] + 1]--;

}

// 累加长度为1到s\_length的本质不同的子串数目

for (int i = 1; i <= s\_length; i++) {

c[i] += c[i - 1];

answer += c[i];

}

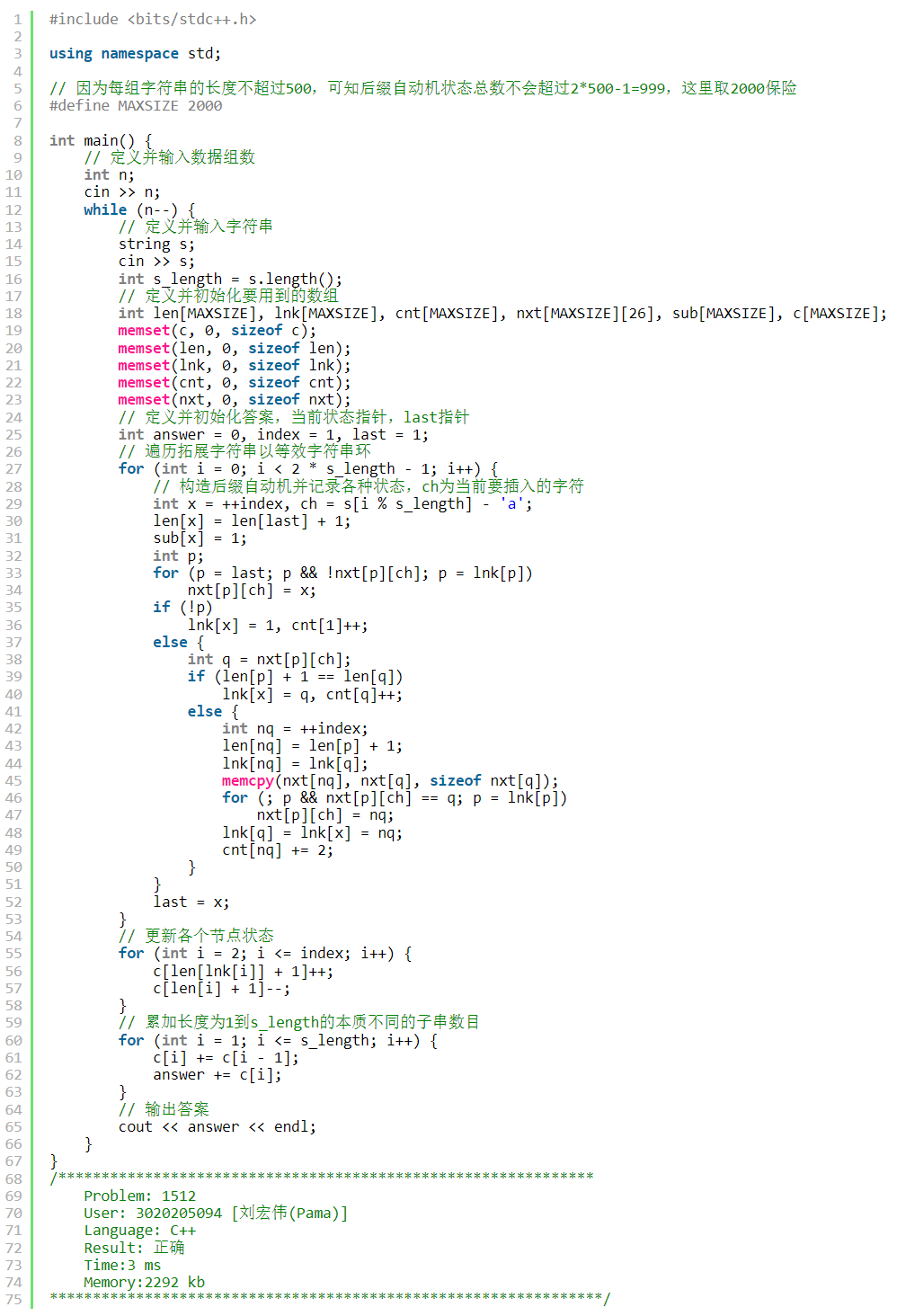
// 输出答案

cout << answer << endl;

}

}

1. **结果展示**（Oj平台提交记录）



# 三．实验中遇到的问题和解决办法

理解后缀自动机的思想和各种资料中五花八门的实现这块花了许多时间，并且将其几点重要性质转换到求本质不同的子串这块也花了许多时间，解决方法是广泛学习，坚持弄懂。