# 2015年厦门大学ACM寒假集训八（0223）

这次我们学学**KMP算法**和**二叉排序树**吧。

**KMP算法**：这个是AC自动机的基础，学计算机的会有一门编译原理的课，这门课机会涉及到各种自动机的概念、其理论和AC自动机差不多，所以学这个至少对学习很有用！

KMP算法并不容易理解，那个next(或者说fail)数组很巧妙，大家要耐心学。

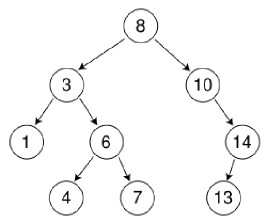
我当时在这里学的KMP算法： <http://www.matrix67.com/blog/archives/115>

**Trie树：**这个很自然的一个数据结构，大家百度理解下就行。

**AC自动机**：AC自动机就是把线性的KMP算法推广到树形KMP算法（用了Trie树的结构），为了减少负担，我们就不学了，最好有时间看看（反正编译原理也要学自动机）。

**二叉排序树** ：每个节点有一个值，左子树上的点的值都比它小，右子树上点的值都比它大。

看看这张图。



我们考虑下如果按顺序插入了1，2，3，4，5....会怎么样？退化成了链是不是。这就使得这种结构很容易被题目的数据针对。

**Treap平衡树** ：对二叉排序树进行了随机化，使数据无法针对。我们也先不看。

这次题目比较少。

Problem 1 : 看懂了KMP就会写。

剪花布条

**Time Limit: 1000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 9960    Accepted Submission(s): 6435**

**Problem Description**

一块花布条，里面有些图案，另有一块直接可用的小饰条，里面也有一些图案。对于给定的花布条和小饰条，计算一下能从花布条中尽可能剪出几块小饰条来呢？

**Input**

输入中含有一些数据，分别是成对出现的花布条和小饰条，其布条都是用可见ASCII字符表示的，可见的ASCII字符有多少个，布条的花纹也有多少种花样。花纹条和小饰条不会超过1000个字符长。如果遇见#字符，则不再进行工作。

**Output**

输出能从花纹布中剪出的最多小饰条个数，如果一块都没有，那就老老实实输出0，每个结果之间应换行。

**Sample Input**

abcde a3

aaaaaa aa

#

**Sample Output**

0

3

**Problem 2** 要对next数组有深入的理解。

Cyclic Nacklace

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 3218    Accepted Submission(s): 1467**

**Problem Description**

给你一个字符串，要求将字符串的全部字符最少循环2次需要添加的字符数。

abcabc 已经循环2次，添加数为0

abcac 没有循环2次，添加字符abcac。数目为5.

abcabcab 已经循环过2次，但第三次不完整，需要添加数为1

**Input**

The first line of the input is a single integer T ( 0 < T <= 100 ) which means the number of test cases.  
Each test case contains only one line describe the original ordinary chain to be remade. Each character in the string stands for one pearl and there are 26 kinds of pearls being described by 'a' ~'z' characters. The length of the string Len: ( 3 <= Len <= 100000 ).

**Output**

For each case, you are required to output the minimum count of pearls added to make a CharmBracelet.

**Sample Input**

3

aaa

abca

abcde

**Sample Output**

0

2

5

**Problem 3**

考虑到KMP并不容易理解，我们就用二叉排序树简单对数组进行排序吧。

问题描述：

这里就希望大家用二叉排序树的思想对输入进行排序吧。

Input

5 1 3 2 4

Output

1 2 3 4 5

可以预见的是：

1、假如输入的数据是随机的，那么排序时间复杂度是O(nlogn)，能应付10^5~10^6的数据。

2、可是如果输入的数据是已经排序号的树，那么时间复杂度变成了O(n\*n)。

那学这个还有用吗？有，这是Treap平衡树的基础。

如果感觉so easy，大家就自己学学treap，可以理解为是对二叉排序树的加强。

Treap会了，可以看看splay树。