**【专辑】AC自动机**

<http://www.cs.uku.fi/~kilpelai/BSA05/lectures/slides04.pdf>  
<http://www.docin.com/p-46845432.html>(上边原文地址如果无法访问的话可以访问这一个~)  
这是我找到的AC自动机最好的资料  
感觉网上其他一些资料都没能阐述的很好，只是草草的介绍一下fail指针，画几张草图而已，贴一下不是很搞笑甚至还带有错误的模板，真正精髓的地方都很模糊的一笔带过  
这片资料虽然是英文的，阅读起来也许有一点点小障碍，而且不像其他一样长篇大论，只是精短的几句描述，但看完之后会有“大彻大悟”的感觉~好了，自己去欣赏这篇极品论文吧..

下边给出几道题目以及解题思路供大家练习  
  
模板题：

[Keywords Search](http://acm.hdu.edu.cn/showproblem.php?pid=2222)

网络流上流传最广的AC自动机模板题，问你目标串中出现了几个模式串

如果一个结点是单词末尾的话out标记为true,在search的时候对于每个结点都向fail指针找，找到out为true的就将其标记为false,且ans++

[病毒侵袭](http://acm.hdu.edu.cn/showproblem.php?pid=2896)

问你目标串中出现了几个模式串

同上一题，只是这题要输出模式串的ID，且字符是所有可见字符,要开[128]的儿子结点，还好没有太卡内存

[病毒侵袭持续中](http://acm.hdu.edu.cn/showproblem.php?pid=3065)

目标串中各个模式串出现了几次

这就更简单了，都不用把out标记成false了~

AC自动机+矩阵

[DNA Sequence](http://acm.pku.edu.cn/JudgeOnline/problem?id=2778)

问你长度为N的串中不包含了模式串的串有几个

n属于1 ~ 2000000000看到这个数据范围我们就应该敏感的想到这是矩阵~  
最多100个结点，先建好所有结点(不包括模式串结尾的和fail指向结尾的结点,所以其实最多只有90个有效结点)之间的转化关系，然后二分矩阵乘法，复杂度O(100^3\*log(2000000000))

[考 研路茫茫——单词情结](http://acm.hdu.edu.cn/showproblem.php?pid=2243)

问你长度为1~N的串中包含了模式串的串总共有几个

上题的加强版，先要把总数26^1 + 26^2 + … + 26^N算出来，然后减去所有不包含的…反正比上题恶心一点点  
答案要模2^64，直接用unsinged \_\_int64 就OK了

AC自动机+DP

[Censored!](http://acm.pku.edu.cn/JudgeOnline/problem?id=1625)

大数+简单DP

[Wireless Password](http://acm.hdu.edu.cn/showproblem.php?pid=2825)

位压缩，每个节点有2^10次状态，找到至少K个

[Ring](http://acm.hdu.edu.cn/showproblem.php?pid=2296)

每个单词有val，关键需要输出路径，比较挺麻烦的，需要倒过来，如果暴力点就直接用string吧。。

[DNA REPAIR](http://acm.hdu.edu.cn/showproblem.php?pid=2457)

就是不要走到尾结点上，如果和匹配串不相同的话+1dp

[Searching the String](http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=3228)

找可以重叠和不能重叠的串个有多少，记录每个单词最后出现的pos，注意有重复的串

[Lost’s revenge](http://acm.hdu.edu.cn/showproblem.php?pid=3341)

RE的神题，非常卡内存。。。空间复杂度(500\*11^4)(这还是缩水后的，原来的是(2500\*11^4))，时间复杂度要再乘上转化复杂度(4)，如果状态变化的系数高一点就过不了。。表示要非常优化的代码才能过

[空罐 Cans](http://zerojudge.tw/ShowProblem?problemid=b179)

高中生题，做了一个小时，关键是debug一个白痴错误de了半个小时。惭愧啊。。不过这题比较好玩的是罐子基因分裂后第一个字符会消失，需要用fail指针来转移状态了  
fail用来转移变短的状态，chd来转移变长的基因，同时还要记录基因的长度(长度l转移的时候需分三类讨论，l = 1，l 恰为根结点到当前结点的距离，l 大于该距离)，DP状态是100\*1500[长度,结点]用滚动数组

[Resource Archiver](http://acm.hdu.edu.cn/showproblem.php?pid=3247)

乳鸽的神题，状态很容易看出来，有50000\*1024，很难保持，我用散列表超时了，用bitset刚好可以卡过，不过后来我想，只有尾结点才有效，中间的很多结点完全可以忽略，可以先用最短路吧各个尾结点之间的距离算出来，经过测试，不到50个点，马上就优化到50\*1024了，本来9s多过的优化到了100多MS

最后献上我的模板

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | #define FF(i,a) for( int i = 0 ; i < a ; i ++ )  #define CC(m,what) memset(m,what,sizeof(m))  const int NODE = 1500;  const int CH = 4;  int chd[NODE][CH] , sz;//结点个数  int word[NODE];//关键数组，记录每个单词尾结点的信息，每道题目都不一样  int fail[NODE];//传说中的失败指针  int Que[NODE];//辅助队列  int sw[128];//string swap每个字符对应的Index，方便模板化    void Ins(char \*a, int val) {  int p = 0;  for(; \*a ; a ++) {  int c = sw[\*a];  if(!chd[p][c]) {  CC(chd[sz] , 0);  word[sz] = 0;  chd[p][c] = sz ++;  }  p = chd[p][c];  }  word[p] = val;  }  void AC() {  int \*s = Que , \*e = Que;  FF(i,CH) if(chd[0][i]) {  fail[ chd[0][i] ] = 0;  \*e++ = chd[0][i];  }  while(s != e) {  int p = \*s++;  FF(i,CH) {  if(chd[p][i]) {  int v = chd[p][i];  \*e++ = v;  fail[v] = chd[fail[p]][i];  //对word[v]按word[fail[v]]里的内容进行处理  } else {  chd[p][i] = chd[fail[p]][i];  }  }  }  }  //AC()函数处理后 chd[p][i] 就是在p结点进行i转移到达的结点  int main() {  fail[0] = 0;  FF(i,26) sw[i+'a'] = i;  //下面两句每次都必须初始化  CC(chd[0],0);  sz = 1;  } |