字符串算法选讲

l0nl1f3

福州第三中学

2017年6月

Tips

- 忘了Tips要写什么了

KMP

- 给定一个长度为n字符串S,给定一个长度为m字符串T,问T在S中出现了几次

KMP

- 给定一个长度为n字符串S,给定一个长度为m字符串T,问T在S中出现了几次
- $n \leq m \leq 5 * 10^6$

KMP

- 朴素匹配,枚举起点l,比较$s_{l...l+m-1}$和T是否相等,遇到不等(失配)就移动l
- 复杂度$O(nm)$

KMP

- 朴素匹配,枚举起点l,比较$s_{l...l+m-1}$和T是否相等,遇到不等(失配)就移动l
- 复杂度$O(nm)$
- KMP(Knuth–Morris–Pratt algorithm),俗称看毛片算法

KMP

- 朴素匹配,枚举起点l,比较$s_{l\ldots l+m-1}$和T是否相等,遇到不等(失配)就移动l
- 复杂度$O(nm)$
- KMP(Knuth–Morris–Pratt algorithm),俗称看毛片算法
- 定义一个串的border为满足$s_{1\ldots x} = s_{n-x+1\ldots n}$的前缀

KMP

- 朴素匹配,枚举起点l,比较$s_{l\ldots l+m-1}$和T是否相等,遇到不等(失配)就移动l
- 复杂度$O(nm)$
- KMP(Knuth–Morris–Pratt algorithm),俗称看毛片算法
- 定义一个串的border为满足$s_{1\ldots x} = s_{n-x+1\ldots n}$的前缀
- KMP的next数组存储的是T串每个前缀的最大border的长度

KMP

- 失配时按border跳

KMP

- 失配时按border跳
- 构建next数组$O(n)$，匹配总复杂度$O(n)$

KMP

- 给定一个长度为n字符串S,给定一个长度为m字符串T,问T在S中出现了几次

KMP

- 给定一个长度为n字符串S,给定一个长度为m字符串T,问T在S中出现了几次
- $n \leq m \leq 5 * 10^6$

Easy Period Problem

- 对于一个字符串T,如果存在字符串A,使得$A + A + \ldots (x*A) = T$,其中加法为顺次连接
- 则称$T = A^x$,求满足条件的最短A的长度

Easy Period Problem

- 对于一个字符串T,如果存在字符串A,使得$A + A + \ldots (x*A) = T$,其中加法为顺次连接
- 则称$T = A^x$,求满足条件的最短A的长度
- 设$T = n - next_n$,若$n \bmod T = 0$,则答案为T
- 正确性如何?

Trie

- 给定n个串,第i个串长为l_i,Q次询问(x, y)的最长公共前缀
- $\sum l_i \leq 5 * 10^6, Q \leq 10^6, n \leq 10^5$

Trie

- 给定n个串,第i个串长为l_i,Q次询问(x, y)的最长公共前缀
- $\sum l_i \leq 5 * 10^6, Q \leq 10^6, n \leq 10^5$
- 建立一棵26叉前缀树(trie).每条边上有一个字母
- 根到某个点的路径组成一个前缀
- 每次插入一个字符串时，就在trie树上把对应路径“填满”，并记录末字符所在的节点

Trie

- 给定n个串,第i个串长为l_i,Q次询问(x, y)的最长公共前缀
- $\sum l_i \leq 5 * 10^6, Q \leq 10^6, n \leq 10^5$
- 建立一棵26叉前缀树(trie).每条边上有一个字母
- 根到某个点的路径组成一个前缀
- 每次插入一个字符串时，就在trie树上把对应路径“填满”，并记录末字符所在的节点
- 将公共前缀查询转化为LCA查询
- $O(\sum_{l_i} \log \sum l_i + Q)$

Trie + KMP

- 给定n个串,第i个串长为l_i
- 再给定一个长度为m的文本串,问有多少个串在文本串中出现过
- $\sum l_i \leq 5 * 10^5, m \leq 10^6, n \leq 10^4$

Trie + KMP

- 给定n个串,第i个串长为l_i
- 再给定一个长度为m的文本串,问有多少个串在文本串中出现过
- $\sum l_i \leq 5 * 10^5, m \leq 10^6, n \leq 10^4$
- 大力KMP,复杂度$O(nm)$

Trie + KMP

- Trie + KMP = Aho-Corasick Automation

Trie + KMP

- Trie + KMP = Aho-Corasick Automation
- 对于Trie上的每一个前缀，我们任然可以建立一个像kmp一样的next数组
- 对于Trie进行bfs，考虑将要遍历i节点的c孩子

Trie + KMP

- Trie + KMP = Aho-Corasick Automation
- 对于Trie上的每一个前缀，我们任然可以建立一个像kmp一样的next数组
- 对于Trie进行bfs，考虑将要遍历i节点的c孩子
- 若c孩子不存在,则将孩子指针指向$next_i$的c孩子
- 否则将该孩子的$next$指向$next_i$节点的c孩子

Trie + KMP

- 那么我们对文本串进行一个Trie上的kmp即可
- $O(\sum l_i + m)$
- HDU2222

Trie + KMP - Trie

- 给出n个字符串,询问每个字符串在所有字符串中出现的次数之和
- $n \leq 10^5, \sum l_i \leq 10^6$

Trie + KMP - Trie

- 给出n个字符串,询问每个字符串在所有字符串中出现的次数之和
- $n \leq 10^5, \sum l_i \leq 10^6$
- 暴力AC自动机的复杂度???

Trie + KMP - Trie

- 给出n个字符串,询问每个字符串在所有字符串中出现的次数之和
- $n \leq 10^5, \sum l_i \leq 10^6$
- 暴力AC自动机的复杂度???
- $O((\sum l_i)^2)$

Trie + KMP - Trie

- 给出n个字符串,询问每个字符串在所有字符串中出现的次数之和
- $n \leq 10^5, \sum l_i \leq 10^6$
- 暴力AC自动机的复杂度???
- $O((\sum l_i)^2)$
- 我们建立一棵fail树,我们把每个节点x向$next_x$连边

Trie + KMP - Trie

- 给出n个字符串,询问每个字符串在所有字符串中出现的次数之和
- $n \leq 10^5, \sum l_i \leq 10^6$
- 暴力AC自动机的复杂度？？？
- $O((\sum l_i)^2)$
- 我们建立一棵fail树,我们把每个节点x向$next_x$连边
- 每个点都是一个字符串的前缀，而且每个字符串的每个前缀在这棵树上都对应着一个点。
- 其次，由于fail指针，每个点父节点的字符串都是这个点字符串的后缀，并且树上没有更长的它的后缀