Lab 4 串匹配算法

实验环境

vlab.ustc.edu.cn
ubuntu 18.04

架构：

```
├── doc
│   ├── report.pdf
├── ex1
│   ├── input
│   │   └── 4_1_input.txt
│   ├── output
│   │   ├── result.txt
│   │   └── time.txt
│   └── src
│       ├── IO.cpp
│       ├── IO.h
│       ├── define.h
│       ├── kmp.cpp
│       ├── kmp.h
│       ├── main.cpp
│       ├── main.h
│       ├── str.cpp
│       ├── str.h
│       ├── timing.cpp
│       └── timing.h
└── ex2
    ├── input
    │   └── 4_2_input.txt
    ├── output
    │   ├── result.txt
    │   └── time.txt
    └── src
        ├── IO.cpp
        ├── IO.h
        ├── d q.txt
        ├── define.h
        ├── main.cpp
        ├── main.h
        ├── rabinkarp.cpp
        ├── rabinkarp.h
        ├── str.cpp
        ├── str.h
        ├── timing.cpp
        └── timing.h
```

实验一:

实验内容:

KMP算法
分析运行时间

实验代码和讲解

与书上伪代码别无二致

```
void kmpMatcher(char t[], char p[]){
    int n = strlen(t);
    int m = strlen(p);
    computePrefixFunction(p);
    int q = 0;
    for(int i = opt.n = 0; i < n; ++i){
        while (q > 0 && p[q] != t[i])
            q = pi[q-1];
        if(p[q] == t[i])
            q++;
        if(q == m) {
            opt.sta[opt.n++] = i - m + 2;    // 记录第n个匹配成功的位置
            q = pi[q-1];
        }
    }
}
```

运行结果

```
input:
BAABABBA
AABABBABBABBBBBABABBABABAAAAAABBBBABABBBBABBBBAABABBBBAAABBAAABBABAAABABAABABBABAABBABBAABBAABABBBABBABBBBBBBABAA

result:
2
0 0 0 1 2 1 1 2
72 189

time:
4
```

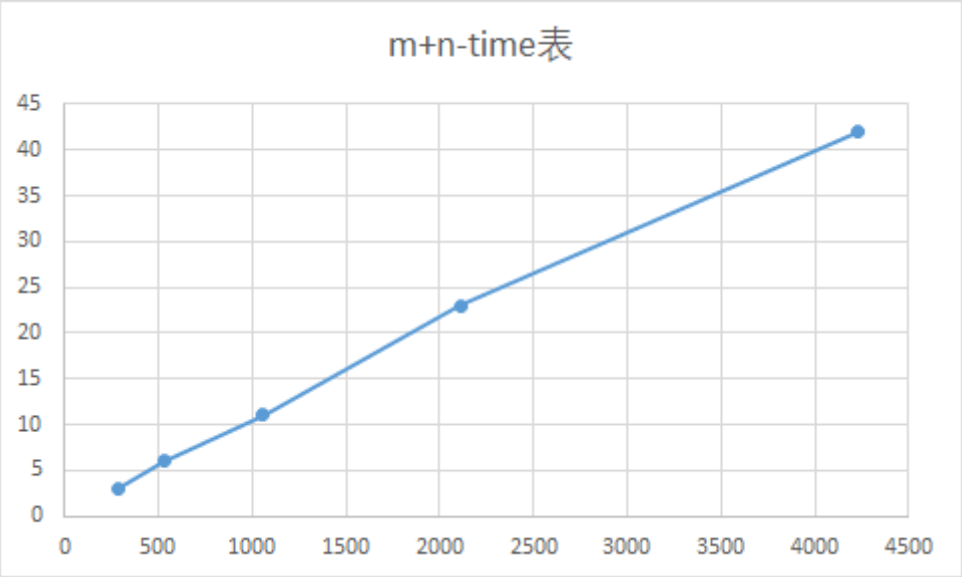
算法复杂度分析

KMP算法运行时间复杂度为O(n+m)

算法运行时间

m+n	time	(m+n)/time
288	3	96
528	6	88

m+n	time	(m+n)/time
1056	11	96
2112	23	91.826
4224	42	100.571



横坐标为m+n,纵坐标为time

> 总结：算法复杂度理论与实际相近

实验二

实验内容:

Rabin-Karp算法

实验代码与讲解

与书上伪代码别无二致

```
void RabinKarp(char T[], char P[],int d, int q){
    int n = strlen(T);
    int m = strlen(P);
    int h = 1; for(int i = 0; i < m-1;++i) h = h*d%q;   // h = d^m-1 mod q
    int p = 0;
    int t = 0;
    for(int i = 0; i < m; ++i){
        p = (d*p+P[i])%q;
        t = (d*t+T[i])%q;
    }
    for(int s = opt.f[opt.i] = opt.n = 0; s <= n-m; ++s){
        if(p == t){
            opt.f[opt.i]++;//命中
            if(!strncmp(P+0,T+s,m))
                opt.sta[opt.n++] = s + 1;   // 真命中
        }
        if(s < n-m){
            t = (d*(t-T[s]*h)+T[s+m]) % q;
            t = (t+q)%q;        // 加上这一行，使得t的值变为非负数
        }
    }
}
```

运行结果与时间分析

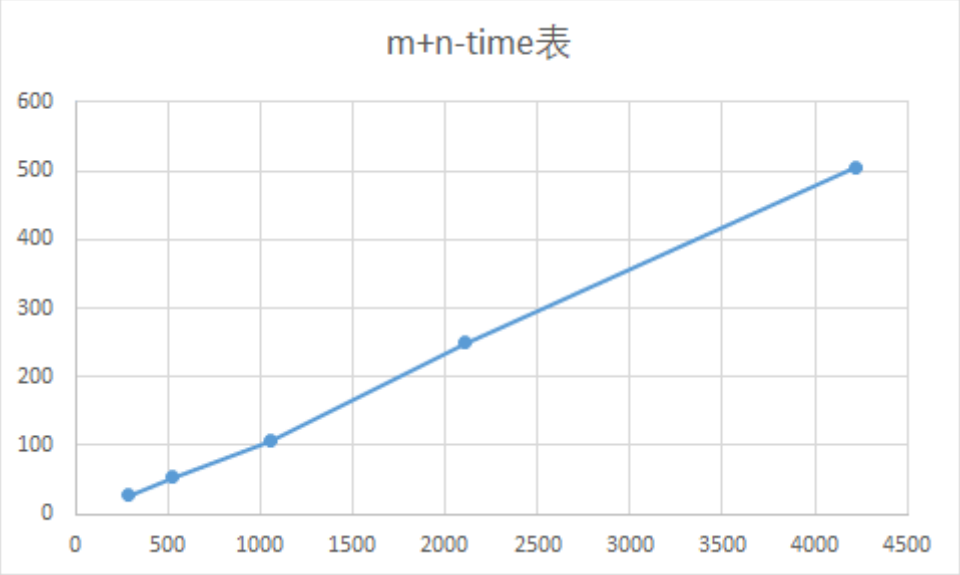
```
input:
z2Xsc7CO
yxBPVmcy4xA2vkGnmGeXudZJjzKnCWo3AAFh8WihqomMKFSVEr5wTIShl40COFbDCTDdzOu9qASxKoz2Xsc7CORsKFreMSnQBJ5tHGhm1H5oQkJ7C


result:
2
15 0 15 0
79 197

time:
27
```

m+n	time	(m+n)/time
288	29	9.93
528	54	9.77
1056	106	9.96
2112	249	8.48

m+n	time	(m+n)/time
4224	506	8.34



横坐标为m+n,纵坐标为time

> 总结：算法复杂度理论与实际相近