

[报告] B - Crazy Search

[Source]

<http://poj.org/problem?id=1200>

[Description]

给出一字符串，已知所含字符种类数为 NC。求长度为 N 的、不同的子串的个数。

[Solution]

对于一个最多可能有 NC 种字符组成的字符串 s (长度为 N)，可以用一个 NC 进制的 N 位数 H 来惟一地表示它。方法是先让每种字符 c 唯一对应一个小于 NC 的数 P[c] ($0 \leq P[c] \leq NC-1$)。于是有 $H = P[s[0]] * NC^{(N-1)} + P[s[1]] * NC^{(N-2)} + \dots + P[s[N-2]] * NC^1 + P[s[N-1]] * NC^0$ 。那么只需要将主字符串扫描一遍，在处理过第一个长度为 N 的子串后，每访问一个字符 s[i] 就更新一下 H ($H = (H - P[s[i-N]] * NC^{(N-1)}) * NC + P[s[i]]$)。开一个 vis 数组记录 H 的值是否已经出现过，统计新的 H 值的个数即可。注意题目没有给明字符类型的范围，不一定只含字母。

[Code]

```
#include<cstdio>
#include<cstring>
using namespace std;
const int MAX = 16e6+5;
int N, NC, P[1<<8];
char s[MAX];
bool vis1[1<<8], vis2[MAX];
int main()
{
    scanf("%d%d", &N, &NC);
    getchar();
    gets(s);
    int len = strlen(s);
    int cnt = 0;
    for (int i = 0; i < len; i++)
        if (!vis1[s[i]])
        {
            vis1[s[i]] = 1;
            P[s[i]] = cnt++;
        }
    int h = 1;
    for (int i = 1; i < N; i++)
        h *= NC;
    int H = 0;
    for (int i = 0; i < N; i++)
        H = NC*H+P[s[i]];
    int ans = 0;
```

```
for (int i = N; ; i++)
{
    if (!vis2[H])
        vis2[H] = 1, ans++;
    if (i == len) break;
    H = NC*(H-h*P[s[i-N]])+P[s[i]];
}
printf("%d\n", ans);
return 0;
}
```