

子串

一. 考察内容:

动态规划

二. 题目分析:

[题目大意]

给出两个字符串a和b，让从a中找出k个子串，顺序拼接之后与b相等，求出方案数。

[写题思路]

考虑用动态规划实现，设状态 $f[i][j][k][0]$:a中前i个字符使用j个字符，这j个字符分布在k个子串，且不使用最后一个字符的方案数， $f[i][j][1]$:a中前i个字符使用j个字符，这j个字符分布在k个子串，且使用最后一个字符的方案数，转移： $f[i][j][k][0]=f[i-1][j][k][0]+f[i-1][j][k][1]$ ， $f[i][j][k][1]=f[i-1][j-1][k][1]$ (将最后一个字符拼到前一个子串末尾)+ $f[i-1][j-1][k-1][1]+f[i-1][j-1][k-1][0]$ (将最后一个字符单独放)。

由于可能会爆内存，考虑把第一维优化掉，用01背包的优化方法，将j、k两维倒序枚举即可。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 17
*文件类型: 源代码文件
*题目来源: COGS
*当前状态: 已通过
*备忘录: 动态规划
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 1001
#define MAXM 202
#define MOD 1000000007
int n,m,K,f[MAXN][MAXM][2];
char a[MAXN],b[MAXM];
/* Variable explain:

*/
void read()
{
    freopen("2015substring.in","r",stdin);
    freopen("2015substring.out","w",stdout);
    scanf("%d%d%d%s", &n,&m,&K,a+1,b+1);
    return;
}
int add(int a,int b)
{
    return (a+b)%MOD;
}
```

```

void dp()
{
    //设状态: f[i][j][k][0]:前i个字母中取出j个分为k组, 其中一定不选第i个的方案数
    //设状态: f[i][j][k][1]:前i个字母中取出j个分为k组, 其中一定选择第i个的方案数
    //第一维状态可被省去
    for(int i=1;i<=n;++i)
    {
        for(int j=m;j;--j)
        {
            for(int k=min(K,j);k;--k)
            {
                f[j][k][0]=add(f[j][k][1],f[j][k][0]);
                if(a[i]==b[j])
                {
                    if(j==1&&a[i]==b[1])f[j][k][1]=1;
                    else f[j][k][1]=add(f[j-1][k][1],add(f[j-1][k-1][1],f[j-1][k-1]
[0]));
                }
                else f[j][k][1]=0;
                // printf("%d %d %d %d\n",i,j,k,f[j][k][1]);
            }
        }
        printf("%d\n",add(f[m][K][0],f[m][K][1]));
    }
}

int main()
{
    read();
    dp();
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)