F-题解

这是一道典型的动态规划问题，**字串**和**子序列**的区别是：子串只能删除前缀或者后缀，而子序列可以删除任意位置的字符。

对于子序列，可以从两个字符串的第0位开始递推，如果两个字符相等，则`f[i+1][j+1]=f[i][j]+1`，否则`f[i+1][j+1]=max(f[i][j+1],f[i+1][j])`。

对于子串，可以从两个字符串的第1位开始递推，如果两个字符相等，则`ff[i+1][j+1]=ff[i][j]+1`，否则`ff[i+1][j+1]=0`(因为子串不能删除任意位置的字符，所以如果两个字符不相等，则`ff[i+1][j+1]`不得不为0)。

以下是核心部分(dp)的代码：

```
void dp(){
    int n=strlen(s),m=strlen(t);
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(s[i]==t[j])f[i+1][j+1]=f[i][j]+1;
            else{
                f[i+1][j+1]=max(f[i][j+1],f[i+1][j]);
            }
        }
    }
    printf("%d\n",f[n][m]);
}
int ff[2005][2005];

void dp2(){
    int ans=0;
    int n=strlen(s),m=strlen(t);
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(s[i]==t[j]){
                ff[i+1][j+1]=ff[i][j]+1;
                ans=max(ans,ff[i+1][j+1]);
            }
            else{
                ff[i+1][j+1]=0;
            }
        }
    }
    printf("%d ",ans);
}
```