**Sunday算法**

    上文中，我们已经介绍了KMP算法和BM算法，这两个算法在最坏情况下均具有线性的查找时间。但实际上，KMP算法并不比最简单的c库函数strstr()快多少，而BM算法虽然通常比KMP算法快，但BM算法也还不是现有字符串查找算法中最快的算法，本文最后再介绍一种比BM算法更快的查找算法即Sunday算法。

    Sunday算法由Daniel M.Sunday在1990年提出，它的思想跟BM算法很相似：

只不过Sunday算法是从前往后匹配，在匹配失败时关注的是文本串中参加匹配的最末位字符的下一位字符。

如果该字符没有在模式串中出现则直接跳过，即移动位数 = 匹配串长度 + 1；

否则，其移动位数 = 模式串中最右端的该字符到末尾的距离+1。

    下面举个例子说明下Sunday算法。假定现在要在文本串"substring searching algorithm"中查找模式串"search"。

    1. 刚开始时，把模式串与文本串左边对齐：  
substring searching algorithm  
search  
^  
    2. 结果发现在第2个字符处发现不匹配，不匹配时关注文本串中参加匹配的最末位字符的下一位字符，即标粗的字符 i，因为模式串search中并不存在i，所以模式串直接跳过一大片，向右移动位数 = 匹配串长度 + 1 = 6 + 1 = 7，从 i 之后的那个字符（即字符n）开始下一步的匹配，如下图：

substring searching algorithm  
　　　 search  
　　　　^  
    3. 结果第一个字符就不匹配，再看文本串中参加匹配的最末位字符的下一位字符，是'r'，它出现在模式串中的倒数第3位，于是把模式串向右移动3位（r 到模式串末尾的距离 + 1 = 2 + 1 =3），使两个'r'对齐，如下：  
substring searching algorithm  
　　　　  search  
　　　　　　　^

    4. 匹配成功。

    回顾整个过程，我们只移动了两次模式串就找到了匹配位置，缘于Sunday算法每一步的移动量都比较大，效率很高。完。