

1.

```
t      = 'abcdaabccddaba'
next(t) = '01111122345123'
s      = 'XYXYXZXZXYYXZ'
next(s) = '01111231212345'
```

2.

删除b和ac,可以选择将不用删除的字符串提到前面,构建两个指针i,j,指针i用于遍历一次字符串,指针j负责拷贝新字符串内容

1. 遇到b直接跳过
2. 遇到c考虑j的位置是否有a,如果有a则将j退一位
3. 其他情况则需要将i处的字符拷贝到处

j指向字符串开头

i遍历字符串:

 若**i*!=b

 若**j*==a

 若**i*=c

 j--

 否则

 *j=*i

 j+=1

 否则

 *j=*i

 j+=1

遍历完毕,设置*j为字符串尾

3.(1)

充要条件: $\exists k \geq 2m - 1, m \in \text{nextset}[k]$

充分性: 假设存在符合条件的 k , 这说明子串 $S(0, k)$ 长度为 m 的前后缀相同, 而该前缀是 P , 该后缀一定与 P 不交, 也即在 T 中, 因此 P 在文本 T 中出现.

必要性: 设 $k = \min\{i | T(i - m, i) = P\}$, k 处有 m 位前后缀相同 $\Rightarrow m \in \text{nextset}[k]$.

(2) $\exists k \geq 2m - 1, m \in \text{nextset}[k]$ 或者 $m - 1 \in \text{nextset}[m + n - 1] \wedge T[n - 1] = P[m - 1]$

1.

首先有字符串求逆算法是线性时间常数空间的:

对给定的 S 的子串 $S(a, b)$, 遍历 $S(a, a + [(b - a)/2])$, 交换该处和对称处的值, 时间复杂度为 $O(b - a)$, 空间复杂度为 $O(1)$

回到原题, 做三次反转, 分别指向 $S(0, n)$, $S(0, n - i)$, $S(n - i, n)$, 即得到了新字符串, 且线性时间常数空间