

原创

后缀数组

2019-08-17 11:15:04 candy_c 阅读数 144 [更多](#)

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/candy_c/article/details/99690656

题目很简单，就是给出一个字符串，把这个字符串的所有非空后缀从小到大排序后，按顺序输出**后缀的第一个字符**在原串中的位置。

样例

输入样例：

ababa

输出样例：

5 3 1 4 2

解释：

排好序后为：

- 1. a
- 2. aba
- 3. ababa
- 4. ba
- 5. baba

暴力肯定不行啦！
这里我们使用一个叫**后缀数组**的东西

PS:搞懂模版还要感谢大佬zhangjianweivv的博客！

什么是后缀数组？

后缀数组为 $rank[i]$ ，为Suffix array的简写，表示**排名为i**的字符串的 **编号**（PS：这里，我们简称排序后后缀的第一个字符所在的位置为 **编号**）

还有一个玩意儿叫**名次数组**，为 $rank[i]$ ，表示**编号为i**的字符串的 **排名**

可以发现， $rank[i]$ 是 $sa[i]$ 的逆运算，而且我们所需要的答案就是 $sa[i]$

后缀数组怎么求？

这才是真正的步入主题啊。。ps:前面讲辣么多废话

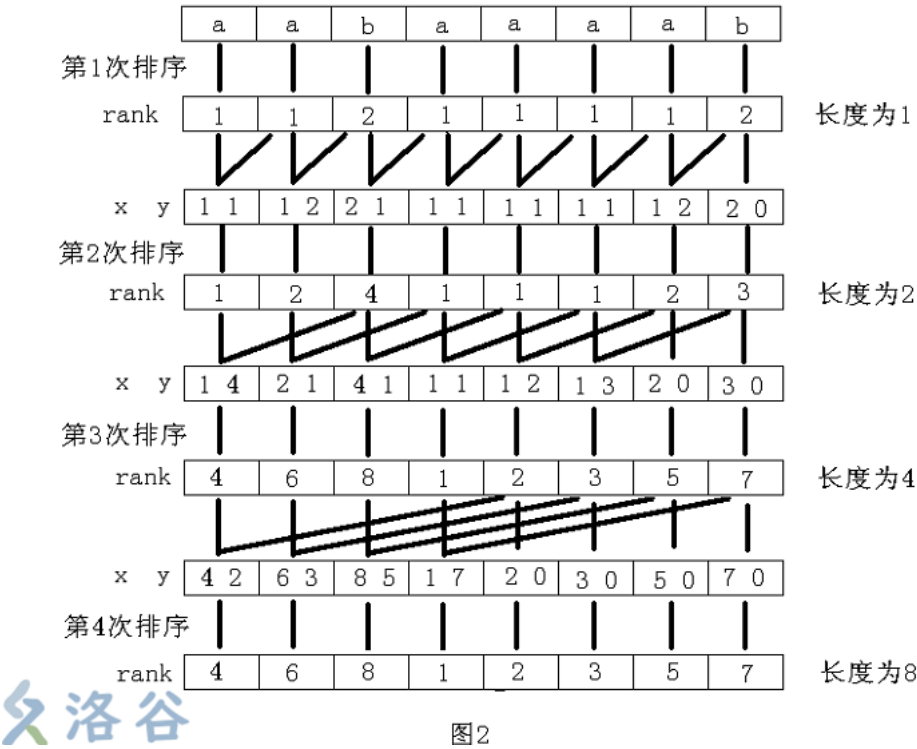
这里我们讲倍增算法，因为好理解我菜

另当前的字符串为 $aabaaaab$

下面这个图很清楚：

如何拿下独角兽公司技术岗

关闭



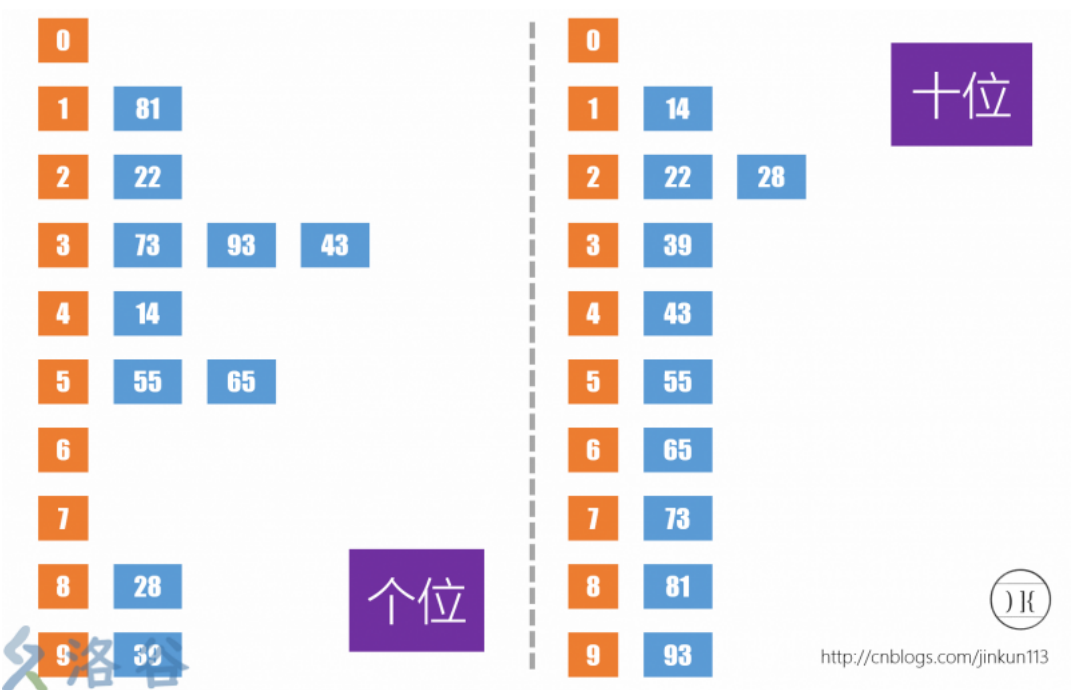
图片来源：《后缀数组—处理字符串有力工具》

- 我们先把字符串的每个字母排序，存在 **rank** 数组里面，我们认为这是**第一关键字**
- 然后将第一关键字每隔 2^k 的字符的排名作为**第二关键字**
- 这时，我们使用**计数排序**（网上很多都说是基数排序，但其实是计数排序）将这两个关键字排序，就类似于两位数字的排序
- 将排序后的结果作为第一关键字，然后继续循环第二，三次操作

计数排序是什么？

计数排序和基数排序很像。

先讲基数排序：



图片来源：jinkun113大佬的博客

类似于桶排序，

- 先按**个位**的数字一个个放入 0_9 的桶里，再按顺序从 0_9 取出来，（当然如果你是从大到小就是 $9\sim 0$ 啦！）
- 接着再按**十位**的数字一个个放入桶里，按顺序取出来，如果是两位数，这时我们得到的就是答案！
- 以此类推，一直循环到最后一位

计数排序，就更高级了。这里优化了一下，

就是我们在基数排序所用的桶，是记录他**出现的次数**，然后**累加**(写入代码就是 `cnt[i]+=cnt[i-1]`)，这样我们得到的 `cnt[i]` 数组里就是**不大于i的个数**名了！

上代码！

我先解释一下每个变量的含义

```

1  rank[i]表示编号为i的排名
2  sa[i]表示排名为i的编号
3  cnt[i]计数排序的桶
4  pos[i]表示当前第二关键字已经排好序时第i名第二关键字所对应的第一关键字位置
5  tmp[i]
6  排序时：表示当前排序中编号为i的排名
7  排序后：表示调整rank前的排名

1  #include<cstdio>
2  #include<cstring>
3  using namespace std;
4  /*
5   rank[i]表示编号为i的排名
6   sa[i]表示排名为i的编号
7   cnt[i]计数排序的桶
8   pos[i]表示当前第二关键字已经排好序时第i名第二关键字所对应的第一关键字位置
9   tmp[i]
10  排序时：表示当前排序中编号为i的排名
11  排序后：表示调整rank前的排名
12  */
13  char s[1200000];
14  int cnt[1200000], pos[1200000], sa[1200000], tmp[1200000], rank[1200000];
15  bool pd(int x, int y, int k){return tmp[x]==tmp[y]&&tmp[x+k]==tmp[y+k];}
16  void suffix(int n, int m)
17  {
18      int i, k;
19      for(i=1; i<=n; i++) rank[i]=s[i], cnt[rank[i]]++;
20      for(i=1; i<=m; i++) cnt[i]+=cnt[i-1];
21      for(i=n; i>=1; i--) sa[cnt[rank[i]]--]=i; // 计数排序，设置好rank和sa数组
22      for(k=1; k<n; k<=1) // k表示长度, k<1表示k*2
23      {
24          int len=0; for(i=n-k+1; i<=n; i++) pos[++len]=i; // 如果第二关键字为0的话，肯定最小，我们先加入pos
25          for(i=1; i<=n; i++) if(sa[i]>k) pos[++len]=sa[i]-k; // 准备好pos数组
26
27          memset(cnt, 0, sizeof(cnt));
28          for(i=1; i<=n; i++) tmp[i]=rank[pos[i]], cnt[tmp[i]]++;
29          for(i=1; i<=m; i++) cnt[i]+=cnt[i-1];
30          for(i=n; i>=1; i--) sa[cnt[tmp[i]]--]=pos[i]; // 更新sa数组
31
32          for(i=1; i<=n; i++) tmp[i]=rank[i]; // 记录之前的排名
33
34          len=1; rank[sa[1]]=1; // 初始化
35          for(i=2; i<=n; i++) if(pd(sa[i], sa[i-1], k)==false) len++; rank[sa[i]]=len; // 通过sa来更新rank数组，并且去重
36          if(len==n) break; m=len;
37      }
38      for(i=1; i<=n; i++) printf("%d ", sa[i]);
39      printf("%d\n", sa[n]);
40  }
41  int main()
42  {
43      scanf("%s", s+1); int len=strlen(s+1);
44      suffix(len, 130); // 'z'是122，为了保险设置为130
45  }

```

```
46 |      return 0;  
    }
```

文章最后发布于: 201

有 0 个人打赏



如何拿下独角兽公司技术岗

关闭