

原创 【LCP】后缀数组 + LCP + RMQ

2019-09-30 19:41:52 我是一只计算鸡 阅读数 15 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/giftedpanda/article/details/101788652>

$LCP(LongestCommonPrefix)$: 最长公共前缀

$rank[i]$: 代表后缀 i 在后缀数组中的下标

$height[i]$: 后缀 $i-1$ 和后缀 i 的最长公共前缀

对于两个后缀 j 和 k ，不妨设 $rank[j] < rank[k]$ ，不难得出后缀 j 和后缀 k 的 LCP 长度等于 $\min(height[rank[j]+1], height[rank[j]+2], \dots, height[rank[k]])$

$LCP(j, k) = RMQ(height, rank[j]+1, rank[k])$

按照定义计算 $height[i]$ 需要 $O(n)$ ，整个 $height$ 数组需要 $O(n^2)$ 。但是我们有 $O(n)$ 的算法，我们用一个辅助数组 $h[i] = height[rank[i]]$ ，然后按照 $h[1]$ 递推计算。递推计算的基于一个性质： $h[i] \geq h[i-1] - 1$ 。

设排在后缀 $i-1$ 前一个是后缀 k 。后缀 k 和后缀 $i-1$ 分别删除首字符之后得到后缀 $k+1$ 和后缀 i ，因此后缀 $k+1$ 一定排在后缀 i 前面，并且最长公共前 $h[i-1]-1$ 。

这个 $h[i-1]-1$ 是一系列 h 值中的最小值，这些 h 值中包括后缀 i 和排在它前一个的后缀 P 的 LCP 长度，即 $h[i]$ 。因此 $h[i] \geq h[i-1] - 1$ 。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 const int maxn = 1000000 + 7;
4 char s[maxn];
5 int sa[maxn], c[maxn], x[maxn], y[maxn], n, m;
6 int Rank[maxn], height[maxn];
7 // LCP rank[i] 后缀 i 的排名, height[i] = LCP(sa[i-1], sa[i])
8 int d[maxn][32]; // d[i][j] 以 i 开始长度为 2^j 的区间的最值
9 int suffixArray() // 后缀数组
10 {
11     for(int i = 1; i <= m; i++) c[i] = 0;
12     for(int i = 1; i <= n; i++) ++c[x[i] = s[i]];
13     for(int i = 2; i <= m; i++) c[i] += c[i-1];
14     for(int i = n; i >= 1; i--) sa[c[x[i]]--] = i;
15     for(int k = 1; k <= n; k <= 1) {
16         int num = 0;
17         for(int i = n - k + 1; i <= n; i++) y[++num] = i;
18         for(int i = 1; i <= n; i++) if(sa[i] > k) y[++num] = sa[i] - k;
19         for(int i = 1; i <= m; i++) c[i] = 0;
20         for(int i = 1; i <= n; i++) ++c[x[i]];
21         for(int i = 2; i <= m; i++) c[i] += c[i-1];
22         for(int i = n; i >= 1; i--) sa[c[x[y[i]]]--] = y[i], y[i] = 0;
23         swap(x, y);
24         num = 1, x[sa[1]] = 1;
25         for(int i = 2; i <= n; i++) {
26             x[sa[i]] = (y[sa[i-1]] == y[sa[i]] && y[sa[i-1] + k] == y[sa[i] + k]) ? num : ++num;
27         }
28         if(num >= n) break;
29         m = num;
30     }
31     return 0;
32 }
33 void getHeight()
34 {
35     int k = 0;
36     // sa[i] 排名为 i 的后缀的下标
37     // Rank[i] 后缀 i 的排名
38     for(int i = 1; i <= n; i++) Rank[sa[i]] = i;
39     for(int i = 1; i <= n; i++) {
40         if(k) k--;
41         int j = sa[Rank[i]-1];
```

```
42 |         while(s[i+k] == s[j+k]) k++; 43 |         height[Rank[i]] = k;
44 |     }
45 | }
46 | void RMQ_init(int* a, int n) // RMQ 初始化
47 | {
48 |     // height数组是从1开始的,而RMQ的数组是从0开始的
49 |     for(int i = 0; i < n; i++) a[i] = a[i+1];
50 |     for(int i = 0; i < n; i++) d[i][0] = a[i];
51 |     for(int j = 1; (1 << j) <= n; j++) {
52 |         for(int i = 0; i + (1 << j) - 1 < n; i++) {
53 |             d[i][j] = min(d[i][j-1], d[i + (1 << (j-1))][j-1]);
54 |         }
55 |     }
56 | }
57 | int RMQ(int l, int r) // RMQ
58 | {
59 |     int k = 0;
60 |     while((1 << (k + 1)) <= r - l + 1) k++;
61 |     return min(d[l][k], d[r-(1<<k)+1][k]);
62 | }
63 | int main()
64 | {
65 |     while(scanf("%s", s + 1) == 1) {
66 |         n = strlen(s + 1);
67 |         m = 122;
68 |         suffixArray();
69 |         getHeight();
70 |         RMQ_init(height, n);
71 |         int l, r;
72 |         scanf("%d %d", &l, &r);
73 |         printf("%d\n", RMQ(l, r));
74 |     }
75 |     return 0;
76 | }
```

有 0 个人打赏

文章最后发布于: 201

©2019 CSDN 皮肤主题: 终极编程指南 设计师: CSDN官方博客