

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

BZOJ5405 platform

2019-01-10 | 题解

首先，我们需要知道如何利用后缀数组对一个指定的字符串计算字典序 Rank。我们以 abaab 为例：

a	a	b			
11	10	9			
a	b				
x	8				
a	b	a	a	b	
x	x	7	6	5	
b					
4					
b	a	a	b		
x	3	2	1		

按照顺序进行编号，因为本质相同的字符串不重复计算次数，所以每一行的前 $height_i$ 个不计入个数，所以我们也有了一个很自然的 $O(n^2)$ 做法。

接下来我们考虑正解，可以发现，每次可以看做对上面的字典序 Rank 数组进行了依次递减的区间覆盖，可以使用线段树维护。同时我们可以发现，这个 Rank 是递减的，而前缀和是递增的，所以我们可以二分出相等的那个位置。

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```
// =====
//  author: memset0
//  date: 2019.01.10 10:37:10
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 4e5 + 10;
int n, m, len, siz;
int a[N], sa[N], rnk[N], tax[N], tmp[N], rank[N], height[N];
char s[N];
ll cnt, sum[N];
std::vector <std::pair <int, int> > ans;

inline ll get_sum(int l, int r) { return sum[r] - sum[l - 1]; }
inline void check_ans(int l, int r, int x) { if (x == sum[r] - sum[l - 1]) an

void get_sa_sort() {
    for (int i = 1; i <= siz; i++) tax[i] = 0;
    for (int i = 1; i <= len; i++) tax[rnk[i]]++;
    for (int i = 1; i <= siz; i++) tax[i] += tax[i - 1];
    for (int i = len; i >= 1; i--) sa[tax[rnk[tmp[i]]]--] = tmp[i];
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

```

}

void get_sa() {
    siz = 300, len = n;
    for (int i = 1; i <= len; i++) rnk[i] = s[i], tmp[i] = i;
    get_sa_sort();
    for (int now = 1, cnt = 0; cnt < len; siz = cnt, now <= 1) {
        cnt = 0;
        for (int i = len; i >= len - now + 1; i--) tmp[++cnt] = i;
        for (int i = 1; i <= len; i++) if (sa[i] > now) tmp[++cnt] = sa[i] -
        get_sa_sort(), std::swap(rnk, tmp), rnk[sa[1]] = cnt = 1;
        for (int i = 2; i <= len; i++) rnk[sa[i]] = (tmp[sa[i]] == tmp[sa[i] -
    }
}

void get_height() {
    for (int i = 1, j, ans = 0; i <= len; i++) {
        j = sa[rnk[i] - 1]; if (ans) --ans;
        while (s[i + ans] == s[j + ans]) ++ans;
        height[rnk[i]] = ans;
    }
    height[1] = 0;
}

struct node { int l, r, mid, siz; ll tag; } p[N << 2];

inline void pushup(int u, ll x) {
    p[u].tag = x;
}

inline void pushdown(int u) {
    if (!p[u].tag || p[u].l == p[u].r) return;
    pushup(u << 1, p[u].tag);
    pushup(u << 1 | 1, p[u].tag - p[u << 1].siz);
    p[u].tag = -1;
}

```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```
void build(int u, int l, int r) {
    p[u].l = l, p[u].r = r, p[u].mid = (l + r) >> 1;
    p[u].tag = -1, p[u].siz = r - l + 1;
    if (l == r) return;
    build(u << 1, l, p[u].mid);
    build(u << 1 | 1, p[u].mid + 1, r);
}

void modify(int u, int l, int r, ll x) {
    pushdown(u);
    if (p[u].l == l && p[u].r == r) return pushup(u, x);
    if (r <= p[u].mid) modify(u << 1, l, r, x);
    else if (l > p[u].mid) modify(u << 1 | 1, l, r, x);
    else modify(u << 1, l, p[u].mid, x), modify(u << 1 | 1, p[u].mid + 1, r, x);
}

ll query(int u, int k) {
    pushdown(u);
    if (p[u].l == p[u].r) return p[u].tag;
    return k <= p[u].mid ? query(u << 1, k) : query(u << 1 | 1, k);
}

void dfs(int u, int l, int r) {
    pushdown(u);
    if (p[u].l == p[u].r) return print(p[u].tag, ' ');
    if (r <= p[u].mid) dfs(u << 1, l, r);
    else if (l > p[u].mid) dfs(u << 1 | 1, l, r);
    else dfs(u << 1, l, p[u].mid), dfs(u << 1 | 1, p[u].mid + 1, r);
}

void main() {
    scanf("%s", s + 1), n = strlen(s + 1), get_sa(), get_height();
    for (int i = 1; i <= n; i++) read(a[i]), sum[i] = sum[i - 1] + a[i];
    build(1, 1, n), cnt = n + (ll)n * (n - 1) / 2;
    for (int i = 1; i <= n; i++) cnt -= height[i];
    ll L, R;
    for (int i = 1; i <= n; i++) {
```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memset0
Proudly powered by Typecho

```

        modify(1, height[i] + 1, n - sa[i] + 1, cnt);
        cnt -= n - sa[i] - height[i] + 1;
        for (int l = sa[i], r = n, mid; l <= r; ) {
            mid = (l + r) >> 1, L = query(1, mid - sa[i] + 1), R = get_sum(sa
            if (L == R) { ans.push_back(std::make_pair(sa[i], mid)); break; }
            else (L > R) ? l = mid + 1 : r = mid - 1;
        }
    }
    std::sort(ans.begin(), ans.end()), print(ans.size(), '\n');
    for (std::vector<std::pair<int, int> > ::iterator it = ans.begin(); it
        print(it->first, ' '), print(it->second, '\n');
}

} signed main() { return ringo::main(), 0; }
```

后缀数组

线段树

用户名

邮箱

网址 (选填)

可以在这里写评论哦 ~



提交评论

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

[BZOJ5404 party](#)
上一篇 «

[BZOJ5403 marshland](#)
» 下一篇

© 2017 - 2019 [memset0 的博客](#).

[浙ICP备19006255号-1](#)

97690 visits · 24756 visitors · 74.48 W words

[关于我](#)
[友情链接](#)
[文章聚合](#)

在这里输入关键字哦 ~ (回车搜索)

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

