## CSES Problem Set

# Substring Order II

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

## Submission details

| | |
|---|---|
| Task: | Substring Order II |
| Sender: | Rodry |
| Submission time: | 2021-12-26 08:29:42 |
| Language: | C++17 |
| Status: | READY |
| Result: | ACCEPTED |

## Test results ▲

| test | verdict | time | |
|---|---|---|---|
| #1 | ACCEPTED | 0.01 s | » |
| #2 | ACCEPTED | 0.01 s | » |
| #3 | ACCEPTED | 0.23 s | » |
| #4 | ACCEPTED | 0.16 s | » |
| #5 | ACCEPTED | 0.33 s | » |
| #6 | ACCEPTED | 0.16 s | » |
| #7 | ACCEPTED | 0.24 s | » |
| #8 | ACCEPTED | 0.56 s | » |

## Code ▲

```
#include<bits/stdc++.h>
using namespace std;

#define int long long
#define endl '\n'
#define F first
```

### String Algorithms

...

| | |
|---|---|
| Counting Patterns | – |
| Pattern Positions | – |
| Distinct Substrings | – |
| Repeating Substring | – |
| String Functions | – |
| Substring Order I | ✓ |
| Substring Order II | ✓ |
| Substring Distribution | ✓ |

### Your submissions

| | |
|---|---|
| 2021-12-26 08:29:42 | ✓ |
| 2021-12-26 08:29:11 | ✓ |
| 2021-12-26 08:29:04 | ✓ |
| 2021-12-26 08:28:30 | ✗ |
| 2021-12-09 22:07:47 | ✗ |

```
#define S second

const int mxN = 1e5+5;
int sa[mxN], pos[mxN], tmp[mxN], lcp[mxN];
int gap, N;
string S;

bool comp(int x, int y) {
    if (pos[x] != pos[y])
        return pos[x] < pos[y];
    x += gap;
    y += gap;
    return (x < N && y < N) ? pos[x] < pos[y] : x > y;
}

void suffix() {
    for (int i = 0; i < N; i++)
        sa[i] = i, pos[i] = S[i];

    for (gap = 1;; gap <<= 1) {
        sort(sa, sa+N, comp);
        for (int i = 0; i < N-1; i++)
            tmp[i+1] = tmp[i] + comp(sa[i], sa[i+1]);
        for (int i = 0; i < N; i++)
            pos[sa[i]] = tmp[i];
        if (tmp[N - 1] == N - 1)
            break;
    }
}

void build_lcp() {
    for (int i = 0, k = 0; i < N; i++) if (pos[i] != N-1) {
        int j = sa[pos[i] + 1];
        while (S[i + k] == S[j + k])
            k++;
        lcp[pos[i]] = k;
        if (k) k--;
    }
}

pair<int,int> seg[mxN*10];
int mark[mxN*10];

void push(int k) {
    if (mark[k]) {
        mark[k] = 0;
        seg[2*k].F = seg[2*k + 1].F = seg[k].F/2;
        seg[2*k].S = seg[2*k + 1].S = 0;
        mark[2*k] = mark[2*k + 1] = 1;
    }
}

void update(int v, int a, int b, int k, int x, int y) {
    if (b < x || a > y) return;
    if (a<=x && b>=y) {
        seg[k].S += v;
        return;
    }
    int h = min(b,y) - max(a,x) + 1;
    push(k);
    seg[k].F += h*v;
    int d = (x+y)/2;
    update(v, a, b, 2*k, x, d);
    update(v, a, b, 2*k + 1, d + 1, y);
}
int assign(int v, int a, int b, int k, int x, int y) {
    if (b < x || a > y) return seg[k].F + (y - x + 1)*seg[k].S;
    if (a <= x && b >= y) {
        seg[k].F = (y-x+1)*v;
        seg[k].S = 0;
        mark[k] = 1;
        return seg[k].F;
    }
    push(k);
    int d = (x+y)/2;
    seg[2*k].S += seg[k].S,  seg[2*k + 1].S += seg[k].S;
    seg[k].S = 0;
    seg[k].F = assign(v, a, b, 2*k, x, d) + assign(v, a, b, 2*k + 1, d + 1, y);
    return seg[k].F;

}
int sum(int a, int b, int k, int x, int y) {
    if (b < x || a > y) return 0;
    if (a <= x && b >= y) {
        return seg[k].F + (y-x+1)*seg[k].S;
    }
    push(k);
    seg[k].F += (y-x+1)*seg[k].S;
    seg[2*k].S += seg[k].S, seg[2*k + 1].S += seg[k].S;
    seg[k].S = 0;
    int d = (x+y)/2;
    return sum(a, b, 2*k, x, d) + sum(a, b, 2*k + 1, d + 1, y);
```

```
}


signed main(){

    cin>>S; N = S.size();
    suffix();
    build_lcp();
    int k; cin>>k;

    k = N*(N+1)/2 - k + 1; //pos from last
    int K = 1<<(int) ceil(log2(N+1));
    int cur = 0;
    for (int i = N-1; i >= 0; i--) {
        update(1, 1, N-sa[i], 1, 0, K-1);
        int prev = (i ? lcp[i-1] : 0);
        int l = prev+1, r = N-sa[i];
        int ans = -1;
        while (l <= r) {
            int m = l + (r-l)/2;
            if (cur + sum(m, N-sa[i], 1, 0, K-1) >= k) {
                ans = m; l = m + 1;
            }
            else r = m - 1;
        }
        if (ans != -1) {
            return cout << S.substr(sa[i], ans), 0;
        }
        cur += sum(prev+1, N-sa[i], 1, 0, K-1);
        assign(0, prev+1, N-sa[i], 1, 0, K-1);
    }
}
```

Share code to others

## Test details ▲

### Test 1

Verdict: ACCEPTED

| input |
|---|
| abaabbaabbab<br>10 |

**correct output**

```
aab
```

**user output**

```
aab
```

### Test 2

Verdict: ACCEPTED

**input**

```
sdmgaasdgiakfatiskwlpswatsgdmu...
```

**correct output**

```
akfatiskwlp
```

**user output**

```
akfatiskwlp
```

### Test 3

Verdict: ACCEPTED

**input**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

**correct output**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

**user output**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

### Test 4

Verdict: ACCEPTED

**input**

```
ababababababababababababababab...
```

**correct output**

```
bababababababababababababababa...
```

**user output**

```
bababababababababababababababa...
```

### Test 5

Verdict: ACCEPTED

**input**

```
bbababaaaaaaabbbabaaaaaabbbaba...
```

**correct output**

```
babaabaababbbbbaababbbaababbaa...
```

**user output**

```
babaabaababbbbbaababbbaababbaa...
```

### Test 6

Verdict: ACCEPTED

**input**

```
xhlqkykuintycceehrvvpquqetdibx...
```

**correct output**

```
vvwpkwzotskdbdwpmejwzbdelqftaw...
```

**user output**

```
vvwpkwzotskdbdwpmejwzbdelqftaw...
```

### Test 7

Verdict: ACCEPTED

**input**

```
qgvqlxktskbljoxnsxvkhvbjupgafe...
```

**correct output**

```
seyyvibngyvlwnxaauhcusdqgvqlxk...
```

**user output**

```
seyyvibngyvlwnxaauhcusdqgvqlxk...
```

### Test 8

Verdict: ACCEPTED

**input**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

**correct output**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

**user output**

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```