



Substring Order II

TASK | SUBMIT | RESULTS | STATISTICS | HACKING

Submission details

Task:	<u>Substring Order II</u>
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	<u>>></u>
#2	ACCEPTED	0.01 s	<u>>></u>
#3	ACCEPTED	0.23 s	<u>>></u>
#4	ACCEPTED	0.16 s	<u>>></u>
#5	ACCEPTED	0.33 s	<u>>></u>
#6	ACCEPTED	0.16 s	<u>>></u>
#7	ACCEPTED	0.24 s	<u>>></u>
#8	ACCEPTED	0.56 s	<u>>></u>

Code ▲

1	#include <bits stdc++.h=""></bits>	1
	using namespace std;	
3		
4	#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

ı		
	2021-12-26 08:29:42	✓
	2021-12-26 08:29:11	✓
	2021-12-26 08:29:04	✓
	2021-12-26 08:28:30	X
I	2021-12-09 22:07:47	X

```
7 #define S second
9 const int mxN = 1e5+5;
10 int sa[mxN], pos[mxN], tmp[mxN], lcp[mxN];
11 int gap, N;
12 string S;
13
   bool comp(int x, int y) {
14
15
       if (pos[x] != pos[y])
16
           return pos[x] < pos[y];</pre>
17
       x += gap;
18
       y += gap;
19
       return (x < N \&\& y < N)? pos[x] < pos[y] : x > y;
20 }
21
22
   void suffix() {
23
       for (int i = 0; i < N; i++)</pre>
           sa[i] = i, pos[i] = S[i];
24
25
26
       for (gap = 1;; gap <<= 1) {</pre>
27
           sort(sa, sa+N, comp);
           for (int i = 0; i < N-1; i++)
28
29
               tmp[i+1] = tmp[i] + comp(sa[i], sa[i+1]);
30
           for (int i = 0; i < N; i++)</pre>
31
                pos[sa[i]] = tmp[i];
32
           if (tmp[N - 1] == N - 1)
33
               break;
34
35 }
36
37
   void build_lcp() {
38
       for (int i = 0, k = 0; i < N; i++) if (pos[i] != N-1) {
39
           int j = sa[pos[i] + 1];
           while (S[i + k] == S[j + k])
40
               k++;
41
42
           lcp[pos[i]] = k;
           if (k) k--;
43
44
45 | }
46
47
   pair<int,int> seg[mxN*10];
   int mark[mxN*10];
48
49
   void push(int k) {
50
51
       if (mark[k]) {
52
           mark[k] = 0;
```

```
seg[2*k].F = seg[2*k + 1].F = seg[k].F/2;
53
           seg[2*k].S = seg[2*k + 1].S = 0;
54
           mark[2*k] = mark[2*k + 1] = 1;
55
56
57 }
58
   void update(int v, int a, int b, int k, int x, int y) {
       if (b < x \mid | a > y) return;
60
61
       if (a<=x && b>=y) {
62
           seg[k].S += v;
63
           return;
64
65
       int h = min(b,y) - max(a,x) + 1;
66
       push(k);
       seg[k].F += h*v;
67
68
       int d = (x+y)/2;
69
       update(v, a, b, 2*k, x, d);
       update(v, a, b, 2*k + 1, d + 1, y);
70
71
72
   int assign(int v, int a, int b, int k, int x, int y) {
       if (b < x \mid | a > y) return seg[k].F + (y - x + 1)*seg[k].S;
73
       if (a <= x \&\& b >= y) {
74
75
           seg[k].F = (y-x+1)*v;
76
           seg[k].S = 0;
77
           mark[k] = 1;
78
           return seg[k].F;
79
       }
       push(k);
80
81
       int d = (x+y)/2;
82
       seg[2*k].S += seg[k].S, seg[2*k + 1].S += seg[k].S;
83
       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);
84
85
       return seg[k].F;
86
87
88 int sum(int a, int b, int k, int x, int y) {
       if (b < x \mid | a > y) return 0;
89
       if (a <= x && b >= y) {
90
           return seg[k].F + (y-x+1)*seg[k].S;
91
92
93
       push(k);
       seg[k].F += (y-x+1)*seg[k].S;
94
       seg[2*k].S += seg[k].S, seg[2*k + 1].S += seg[k].S;
95
       seg[k].S = 0;
96
97
       int d = (x+y)/2;
98
       return sum(a, b, 2*k, x, d) + sum(a, b, 2*k + 1, d + 1, y);
```

```
99 }
100
101
102 | signed main(){
103
        cin>>S; N = S.size();
104
        suffix();
105
106
        build lcp();
107
        int k; cin>>k;
108
        k = N*(N+1)/2 - k + 1; //pos from last
109
110
         int K = 1<<(int) ceil(log2(N+1));</pre>
111
        int cur = 0;
        for (int i = N-1; i >= 0; i--) {
112
             update(1, 1, N-sa[i], 1, 0, K-1);
113
             int prev = (i ? lcp[i-1] : 0);
114
115
             int l = prev+1, r = N-sa[i];
             int ans = -1;
116
117
             while (1 <= r) {
118
                 int m = 1 + (r-1)/2;
                 if (cur + sum(m, N-sa[i], 1, 0, K-1) >= k) {
119
120
                     ans = m; 1 = m + 1;
121
                 else r = m - 1;
122
123
124
            if (ans != -1) {
125
                 return cout << S.substr(sa[i], ans), 0;</pre>
126
127
             cur += sum(prev+1, N-sa[i], 1, 0, K-1);
128
             assign(0, prev+1, N-sa[i], 1, 0, K-1);
129
130 }
```

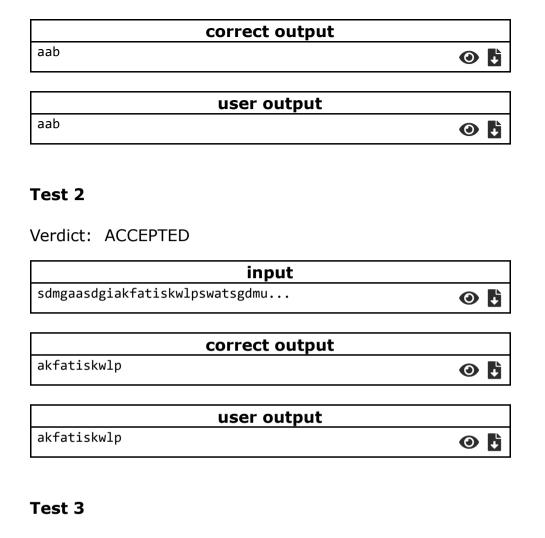
Share code to others

Test details ▲

Test 1

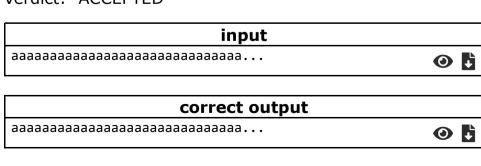
Verdict: ACCEPTED

```
input
abaabbaabbab
10
```



Verdict: ACCEPTED

aaaaaaaaaaaaaaaaaaaaaa...



user output

O

Test 4

Verdict: ACCEPTED

input	
abababababababababababab	©

correct output	
babababababababababababa	0

user output	
babababababababababababa	②

Test 5

Verdict: ACCEPTED

input	
bbababaaaaaabbbabaaaaabbbaba	Ø

correct output	
babaabaababbbaababbaa	O

user output	
babaabaababbbaababbaa	•

Test 6

Verdict: ACCEPTED

input	
xhlqkykuintycceehrvvpquqetdibx	Ø

