CSES Problem Set

Substring Distribution

Submission details

Task:	Substring Distribution
Sender:	seleneal1996
Submission time:	2021-12-15 05:18:25
Language:	C++17
Status:	READY
Result:	ACCEPTED

Test results

test	verdict	time	
#1	ACCEPTED	0.01 s	»
#2	ACCEPTED	0.01 s	»
#3	ACCEPTED	0.12 s	»
#4	ACCEPTED	0.12 s	»
#5	ACCEPTED	0.10 s	»
#6	ACCEPTED	0.12 s	»
#7	ACCEPTED	0.13 s	»
#8	ACCEPTED	0.12 s	»
#9	ACCEPTED	0.13 s	»

Code

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

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]
}

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

    for (gap = 1;; gap <<= 1) {
```

String Algorithms

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

Your submissions

2021-12-15 05:18:25 ✓

```
        sort(sa, sa+N, comp);
        for (int i = 0; i < N-1; i++)
            tmp[i+1] = tmp[i] + comp(sa[i], s
        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 (po
        int j = sa[pos[i] + 1];
        while (S[i + k] == S[j + k])
            k++;
        lcp[pos[i]] = k;
        if (k) k--;
    }
}

int pre[mxN];

int  main(){

    cin>>S; N = S.size();
    suffix();
    build_lcp();
    int prev = 0;
    for (int i = 0; i < N; i++) {
        pre[prev + 1]++;
        pre[N - sa[i] + 1]--;
        prev = lcp[i];
    }
    for (int i = 1; i <= N; i++) {
        cout << pre[i] << ' ' ;
        pre[i+1] += pre[i];
    }

    return 0;
}
```

Share code to others

Test details

Test 1

Verdict: ACCEPTED

input
aaaaaaaaa

correct output
1 1 1 1 1 1 1 1 1

user output
1 1 1 1 1 1 1 1 1

Test 2

Verdict: ACCEPTED

input

```
dfgfglrisgdsdgskd
```

correct output

```
8 15 15 14 13 12 11 10 9 8 7 6...
```

user output

```
8 15 15 14 13 12 11 10 9 8 7 6...
```

Test 3

Verdict: ACCEPTED

input

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

correct output

```
2 3 4 5 6 7 8 9 10 11 12 13 14...
```

user output

```
2 3 4 5 6 7 8 9 10 11 12 13 14...
```

Test 4

Verdict: ACCEPTED

input

```
abababababababababababababab...
```

correct output

```
2 4 8 10 12 14 16 18 20 22 24 ...
```

user output

```
2 4 8 10 12 14 16 18 20 22 24 ...
```

Test 5

Verdict: ACCEPTED

input

abcdabcdabcdabcdabcdabcdabcdab...

correct output

4 16 41 56 73 90 107 125 144 1...

user output

4 16 41 56 73 90 107 125 144 1...

Test 6

Verdict: ACCEPTED

input

mjuhshbeqhmjuhshbeqhmjuhshbeqh...

correct output

8 42 84 114 140 166 192 218 24...

user output

8 42 84 114 140 166 192 218 24...

Test 7

Verdict: ACCEPTED

input

sjhvsnkzzhxnrwrkdhbhjsxrdmazhu...

correct output

25 125 159 180 200 220 240 260...

user output

25 125 159 180 200 220 240 260...

Test 8

Verdict: ACCEPTED

input

wovxwnsrmgvjcvwepxviodztwrkszm...

correct output

26 676 7670 10103 10279 10336 ...

user output

```
26 676 7670 10103 10279 10336 ...
```

Test 9

Verdict: ACCEPTED

input

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa...
```

correct output

```
2 3 4 5 6 7 8 9 10 11 12 13 14...
```

user output

```
2 3 4 5 6 7 8 9 10 11 12 13 14...
```