CCC '20 J4 - Cyclic Shifts (Hard)

Time limit: 0.4s **Memory limit:** 512M
Java: 0.6s
Python: 0.7s

This is an extension of ccc20j4. In this version, the constraints are higher.

Thuc likes finding cyclic shifts of strings. A *cyclic shift* of a string is obtained by moving characters from the beginning of the string to the end of the string. We also consider a string to be a cyclic shift of itself. For example, the cyclic shifts of `ABCDE` are:

`ABCDE`, `BCDEA`, `CDEAB`, `DEABC`, and `EABCD`.

Given some text, T, and a string, S, determine if T contains a cyclic shift of S.

Input Specification

The input will consist of exactly two lines containing only uppercase letters. The first line will be the text T, and the second line will be the string S.

Subtask 1 [90%]

Each line will contain at most $200\,000$ characters.

Subtask 2 [10%]

Each line will contain at most 10^7 characters.

Tip: the intended solution runs well within the time limit without constant optimization.

Output Specification

Output `yes` if the text, T, contains a cyclic shift of the string, S. Otherwise, output `no`.

Sample Input 1

```
ABCCDEABAA
ABCDE
```

Output for Sample Input 1

```
yes
```

Explanation of Output for Sample Input 1

`CDEAB` is a cyclic shift of `ABCDE` and is contained in the text `ABC` **`CDEAB`** `AA`.

Sample Input 2

```
ABCDDEBCAB
ABA
```

Output for Sample Input 2

```
no
```

Explanation of Output for Sample Input 2

The cyclic shifts of `ABA` are `ABA`, `BAA`, and `AAB`. None of these shifts are contained in the text `ABCDDEBCAB`.