6202. Using a Robot to Print the Lexicographically Smallest String

My Submissions (/contest/weekly-contest-314/problems/using-a-robot-to-print-the-lexicographically-smallest-string/submissions/)

Back to Contest (/contest/weekly-contest-314/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

You are given a string `s` and a robot that currently holds an empty string `t`. Apply one of the following operations until `s` and `t` **are both empty**:

- Remove the **first** character of a string `s` and give it to the robot. The robot will append this character to the string `t`.
- Remove the **last** character of a string `t` and give it to the robot. The robot will write this character on paper.

Return *the lexicographically smallest string that can be written on the paper.*

Example 1:

```
Input: s = "zza"
Output: "azz"
Explanation: Let p denote the written string.
Initially p="", s="zza", t="".
Perform first operation three times p="", s="", t="zza".
Perform second operation three times p="azz", s="", t="".
```

Example 2:

```
Input: s = "bac"
Output: "abc"
Explanation: Let p denote the written string.
Perform first operation twice p="", s="c", t="ba".
Perform second operation twice p="ab", s="c", t="".
Perform first operation p="ab", s="", t="c".
Perform second operation p="abc", s="", t="".
```

Example 3:

```
Input: s = "bdda"
Output: "addb"
Explanation: Let p denote the written string.
Initially p="", s="bdda", t="".
Perform first operation four times p="", s="", t="bdda".
Perform second operation four times p="addb", s="", t="".
```

Constraints:

- `1 <= s.length <= 10^5`
- `s` consists of only English lowercase letters.

JavaScript

```
const ord = (c) => c.charCodeAt();
const char = (ascii) => String.fromCharCode(ascii);

const robotWithString = (s) => {
    let f = Array(26).fill(0), t = [], res = '';
    for (const c of s) f[ord(c) - 97]++;
    for (const c of s) {
        f[ord(c) - 97]--;
        t.push(c);
        while (t.length) {
            let find = false;
            for (let i = 0; i < 26; i++) {
                let curC = char(i + 97);
                if (curC < t[t.length - 1]) {
                    if (f[i] > 0) {
                        find = true;
                        break;
```

```
                    }
                }
            }
            if (find) {
                break;
            } else {
                res += t.pop();
            }
        }
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/818378555/) | More Details > (/submissions/detail/818378555/)

Share your acceptance!