

75 Days of Code Day 26 Problem no:394. Decode String
(leetcode)
Type : Stacks

Given an encoded string, return its decoded string.

The encoding rule is: $k[\text{encoded_string}]$, where the `encoded_string` inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer.

You may assume that the input string is always valid; there are no extra white spaces, square brackets are well-formed, etc.
Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k . For example, there will not be input like `3a` or `2[4]`.

The test cases are generated so that the length of the output will never exceed 105.

Example 1:

Input: `s = "3[a]2[bc]"`

Output: `"aaabcbcb"`

Example 2:

Input: `s = "3[a2[c]]"`

Output: `"accaccacc"`

Example 3:

Input: `s = "2[abc]3[cd]ef"`

Output: "abcabccdcddcdef"

Solution using Stack

- Create two stacks number stack and string stack and iterate a loop
- Push all the number in the number stack
- Push all the string in string stack
- If top element === ']' pop all the element until top element === '['
- Multiply all the popped element with number of time the the top element of number stack and pop the top element

```
...  README.md  day26.ts U  day26.js 9+, U  day25.ts  day24.ts  day23.ts
day26.ts > ...
1  function decodeString(s: string): string {
2      let chars = s.split("");
3      let stringsStack = [];
4      let numberStack: number[] = [];
5      let stringPointer = 0;
6      let num = 0;
7
8      for (let index = 0; index < s.length; index++) {
9          if (Number(chars[index]) >= 0 && Number(chars[index]) <= 9) {
10             num = num * 10 + Number(chars[index]) - 0;
11             continue;
12         }
13         if (chars[index] == "[") {
14             numberStack.push(num);
15             num = 0;
16             stringsStack.push(chars[index]);
17             stringPointer++;
18             continue;
19         }
20         if (chars[index] !== "]") {
21             stringsStack.push(chars[index]);
22             stringPointer++;
23             continue;
24         }
25         let repeatString = "";
26         while (
27             stringsStack.length > 0 &&
28             stringsStack[stringsStack.length - 1] !== "["
29         ) {
30             repeatString = "" + stringsStack[stringsStack.length - 1] + repeatString;
31             stringsStack.pop();
32             stringPointer--;
33         }
34
35         stringsStack.pop();
36         let repeatLength: number = Number(numberStack.pop());
37         let resultStr = "";
38         while (repeatLength > 0) {
39             resultStr = repeatString + resultStr;
40             repeatLength--;
41         }
42         stringsStack.push(resultStr);
43     }
44
45     return stringsStack.join("");
46 }
9+, U
U
47
48 let ans1 = decodeString("100[leetcode]");
49 console.log("ans", ans1);
50
```

[illegible]