

Description

Solution

Discuss (999+)

Submissions

Java

Autocomplete

402. Remove K Digits

Medium

4795

202

Add to List

Share

Given string `num` representing a non-negative integer `num` , and an integer `k` , return *the smallest possible integer after removing `k` digits from `num`* .

Example 1:

Input: `num = "1432219"`, `k = 3`
Output: `"1219"`
Explanation: Remove the three digits 4, 3, and 2 to form the new number 1219 which is the smallest.

Example 2:

Input: `num = "10200"`, `k = 1`
Output: `"200"`
Explanation: Remove the leading 1 and the number is 200. Note that the output must not contain leading zeroes.

Example 3:

Input: `num = "10"`, `k = 2`
Output: `"0"`
Explanation: Remove all the digits from the number and it is left with nothing which is 0.

Constraints:

- $1 \leq k \leq \text{num.length} \leq 10^5$
- `num` consists of only digits.
- `num` does not have any leading zeros except for the zero itself.

Accepted 221,068

Submissions 751,294

Seen this question in a real interview before?

Yes

No

Problems

Pick One

< Prev

127

Next >

Console

Contribute

Run Code

Submit

```
1  class Solution {
2
3      public String removeKdigits(String num, int k) {
4
5          int length = num.length();
6
7          if(k >= length)
8              return "0";
9
10         Stack<Character> stack = new Stack<>();
11
12         int i =0;
13
14         while(i < length){
15
16             while(k>0 && !stack.isEmpty() && stack.peek().charAt(i)){
17                 stack.pop();
18                 k--;
19             }
20
21             stack.push(num.charAt(i));
22
23             i++;
24         }
25
26         while(k > 0){
27             stack.pop();
28             k--;
29         }
30
31         StringBuilder sb = new StringBuilder();
32
33         while(!stack.isEmpty())
34             sb.append(stack.pop());
35
36         sb.reverse();
37
38         while(sb.length()>1 && sb.charAt(0)=='0')
39             sb.deleteCharAt(0);
40
41         return sb.toString();
42     }
43 }
```