**[Remove K Digits](https://leetcode.com/problems/remove-k-digits/)**

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 <= k <= num.length <= 105
* num consists of only digits.
* num does not have any leading zeros except for the zero itself.

class Solution {

public:

std::string removeKdigits(std::string num, int k) {

std::stack<char> stack;

for (char digit : num) {

while (!stack.empty() && k > 0 && stack.top() > digit) {

stack.pop();

k--;

}

stack.push(digit);

}

// Remove remaining k digits from the end of the stack

while (k > 0 && !stack.empty()) {

stack.pop();

k--;

}

// Construct the resulting string from the stack

std::string result;

while (!stack.empty()) {

result += stack.top();

stack.pop();

}

std::reverse(result.begin(), result.end()); // Reverse to get the correct order

// Remove leading zeros

size\_t pos = result.find\_first\_not\_of('0');

result = (pos == std::string::npos) ? "0" : result.substr(pos);

return result;

}

};

Link : https://leetcode.com/problems/remove-k-digits/solutions/5005706/faster-lesser-detailed-explaination-stack-greedy-step-by-step-explaination-python-java-c/?envType=daily-question&envId=2024-04-11