Add Binary Strings

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	String Manipulation
language	C++
solved on	@29/11/2024
link	https://www.geeksforgeeks.org/problems/add-binary-strings3805/1
Completion	☑

Intuition

The problem is about adding two binary numbers represented as strings. Binary addition follows the same principle as decimal addition, but with only two digits (`0` and `1`). The carry in binary addition can only be `0` or `1`, making the logic straightforward.

The idea is to traverse both strings from the least significant bit to the most significant bit, add corresponding bits along with any carry from the previous operation, and store the result.

Approach

1. **Two-Pointer Technique:** Use two pointers `i` and `j` to traverse `s1` and `s2` from the end towards the beginning.
2. **Carry Management:** Maintain a `carry` variable initialized to `0` to handle the overflow from the previous sum.
3. **Summation Loop:**
   - Add the `carry`.
   - Add the corresponding bit from `s1` if `i >= 0`.
   - Add the corresponding bit from `s2` if `j >= 0`.
   - Append the least significant bit of the sum (`sum & 1`) to `answer`.
   - Update `carry` to the most significant bit (`sum >> 1`).
4. **Final Carry Check:** If `carry` remains after the loop, append `'1'` to `answer`.
5. **Cleanup:** Remove any leading zeros by popping from the back.
6. **Reversal:** Reverse `answer` since the result is accumulated backward.

Complexity

Time Complexity:

- Each string is traversed once, with length `n` for `s1` and `m` for `s2`.
- Total complexity:**O(max(n, m))**

Space Complexity:

- The space used by `answer` is proportional to the maximum length of the two input strings.**O(max(n, m))**

Code

```
class Solution {
  public:
    string addBinary(string& s1, string& s2) {
        string answer;

        int sum = 0;
        int carry = 0;
        int i = s1.size() - 1;
        int j = s2.size() - 1;
        int k = 0;

        while(i >= 0 || j >= 0){
            sum = carry;
            if(0 <= i) sum += s1[i--] - '0';
            if(0 <= j) sum += s2[j--] - '0';

            answer.push_back((sum & 1) + '0');
            carry = sum >> 1;
        }
        if(carry) answer.push_back('1');

        while(answer.back()=='0') answer.pop_back();
        reverse(answer.begin(), answer.end());

        return answer;

    }
};
```