Add Binary

Given two binary strings, return their sum (also a binary string).

For example,
a = `"11"`
b = `"1"`
Return `"100"`.

Solution 1

```
class Solution
{
public:
    string addBinary(string a, string b)
    {
        string s = "";

        int c = 0, i = a.size() - 1, j = b.size() - 1;
        while(i >= 0 || j >= 0 || c == 1)
        {
            c += i >= 0 ? a[i --] - '0' : 0;
            c += j >= 0 ? b[j --] - '0' : 0;
            s = char(c % 2 + '0') + s;
            c /= 2;
        }

        return s;
    }
};
```

written by makuiyu original link here

Solution 2

```
public class Solution {
    public String addBinary(String a, String b) {
        if(a == null || a.isEmpty()) {
            return b;
        }
        if(b == null || b.isEmpty()) {
            return a;
        }
        char[] aArray = a.toCharArray();
        char[] bArray = b.toCharArray();
        StringBuilder stb = new StringBuilder();

        int i = aArray.length - 1;
        int j = bArray.length - 1;
        int aByte;
        int bByte;
        int carry = 0;
        int result;

        while(i > -1 || j > -1 || carry == 1) {
            aByte = (i > -1) ? Character.getNumericValue(aArray[i--]) : 0;
            bByte = (j > -1) ? Character.getNumericValue(bArray[j--]) : 0;
            result = aByte ^ bByte ^ carry;
            carry = ((aByte + bByte + carry) >= 2) ? 1 : 0;
            stb.append(result);
        }
        return stb.reverse().toString();
    }
}
```

Addition bits are calculated by xor. Carry bit is calculated as simple integer addition.

written by markivr original link here

Solution 3

```
public class Solution {
    public String addBinary(String a, String b) {
        StringBuilder sb = new StringBuilder();
        int i = a.length() - 1, j = b.length() -1, carry = 0;
        while (i >= 0 || j >= 0) {
            int sum = carry;
            if (j >= 0) sum += b.charAt(j--) - '0';
            if (i >= 0) sum += a.charAt(i--) - '0';
            sb.append(sum % 2);
            carry = sum / 2;
        }
        if (carry != 0) sb.append(carry);
        return sb.reverse().toString();
    }
}
```

Computation from string usually can be simplified by using a carry as such.

written by lx223 original link here

From Leetcoder.