

43. Multiply Strings

[link](#)

写错了两次, 一次是0的话, 最好提前判断, 还有就是length最大是两个相加, 我们算的时候实际是反过来的, 注意index别弄混了, 这里借用了LinkedList那里的思想.

```
public String multiply(String num1, String num2) {
    if("0".equals(num1) || "0".equals(num2)) return "0";
    int[] res = new int[num1.length() + num2.length()];
    for(int i = num1.length() - 1; i >= 0; --i){
        for(int j = num2.length() - 1; j >= 0; --j){
            int index = num2.length() - 1 - j + num1.length() - 1 - i;
            res[index] += (num1.charAt(i) - '0') * (num2.charAt(j) - '0');
        }
    }
    StringBuilder sb = new StringBuilder();
    for(int i = 0; i < res.length; ++i){
        int over = res[i] / 10;
        int digit = res[i] % 10;
        sb.append(digit);
        if(over != 0)
            res[i + 1] += over;
    }
    String s = sb.reverse().toString();
    int j = 0;
    while(j < s.length()){
        if(!Character.valueOf(s.charAt(j)).equals('0'))
            break;
        ++j;
    }
    return s.substring(j);
}
```

这个写的更全面[link](#)

```
public String multiply(String num1, String num2) {
    int m = num1.length(), n = num2.length();
    int[] pos = new int[m + n];

    for(int i = m - 1; i >= 0; i--) {
        for(int j = n - 1; j >= 0; j--) {
            int mul = (num1.charAt(i) - '0') * (num2.charAt(j) - '0');
            int p1 = i + j, p2 = i + j + 1;
            int sum = mul + pos[p2];

            pos[p1] += sum / 10;
```

```
        pos[p2] = (sum) % 10;
    }
}

StringBuilder sb = new StringBuilder();
for(int p : pos) if(!(sb.length() == 0 && p == 0)) sb.append(p);
return sb.length() == 0 ? "0" : sb.toString();
}
```