

JS 矩阵最大值

题目描述：给定一个仅包含0和1的N*N二维矩阵，请计算二维矩阵的最大值，计算规则如下：

- 1、每行元素按下标顺序组成一个二进制数（下标越大越排在低位），二进制数的值就是该行的值。矩阵各行值之和为矩阵的值。
- 2、允许通过向左或向右整体循环移动每行元素来改变各元素在行中的位置。
比如：[1,0,1,1,1]向右整体循环移动2位变为[1,1,1,0,1]，二进制数为11101，值为29。
[1,0,1,1,1]向左整体循环移动2位变为[1,1,1,1,0]，二进制数为11110，值为30。

输入描述：1、输入的第一行为正整数，记录了N的大小， $0 < N \leq 20$ 。
2、输入的第2到N+1行为二维矩阵信息，行内元素边角逗号分隔。

输出描述：矩阵的最大值。

示例

示例1

输入：5

1,0,0,0,1
0,0,0,1,1
0,1,0,1,0

1,0,0,1,1
1,0,1,0,1

输出：122

说明：第一行向右整体循环移动1位，得到本行的最大值[1,1,0,0,0]，二进制值为11000，十进制值为24。
第二行向右整体循环移动2位，得到本行的最大值[1,1,0,0,0]，二进制值为11000，十进制值为24。
第三行向左整体循环移动1位，得到本行的最大值[1,0,1,0,0]，二进制值为10100，十进制值为20。
第四行向右整体循环移动2位，得到本行的最大值[1,1,1,0,0]，二进制值为11100，十进制值为28。
第五行向右整体循环移动1位，得到本行的最大值[1,1,0,1,0]，二进制值为11010，十进制值为26。
因此，矩阵的最大值为122。

```
1  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void async function () {
6      const matrix = []
7      let n = 0
8      let i = 0
9      while(line = await readline()){
10         let tokens = line.split(',')
11         if (i === 0) {
12             n = parseInt(tokens[0])
13         }
14         else {
15             let tmp = tokens.map(e => parseInt(e))
16             matrix.push(tmp)
17         }
18         i++
19     }
20
21     console.log(maxOfMatrix(n, matrix))
22 }()
23
24 function maxOfMatrix(n, matrix) {
25     let ans = 0
26     for(let nums of matrix) {
27         let max = 0
28         for(let i = 0; i < n; i++) {
29             max = Math.max(max, binaryToDecimal(nums))
30             nums.push(nums.shift())
31         }
32         ans += max
33     }
34     return ans
35 }
36
37 function binaryToDecimal(arr) {
38     return parseInt(arr.join(""), 2)
39 }
```