

Java-广搜-给你一个由 '0'空地、 '1'银矿

题目描述：

给你一个由 '0'（空地）、 '1'（银矿）、 '2'（金矿）组成的的地图，矿堆只能由上下左右相邻的金矿或银矿连接形成。超出地图范围可以认为是空地。

假设银矿价值 1，金矿价值 2，请你找出地图中最大价值的矿堆并输出该矿堆的价值

输入描述：

地图元素信息如：

22220

00000

00000

11111

地图范围最大 300*300

0<= 地图元素 <=2

输出描述：

矿堆的最大价值

补充说明：

示例 1

输入：

22220

00000

00000

01111

输出：

8

说明：

示例 2

输入：

22220

00020

00010

01111

输出：

15

说明：

示例 3

输入：

20000

00020

00000

00111

输出：

3

说明：

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
```

// 注意类名必须为 Main, 不要有任何 package xxx 信息

```
public class Main {
    public static void buildList(List<List<Integer>> arrays, String str) {
        List<Integer> list = new ArrayList<>();
        for (char c : str.toCharArray()) {
            list.add(c - '0');
        }
        arrays.add(list);
    }
}
```

```
    public static void main(String[] args) {
        List<List<Integer>> arrays;
        int n;
        int m;
        Scanner in = new Scanner(System.in);
        arrays = new ArrayList<>();
        while (in.hasNextLine()) {
            String str = in.nextLine();
            if (str.length() == 0) {
                continue;
            }
            buildList(arrays, str);
        }
        n = arrays.size();
        m = arrays.get(0).size();
        int res = 0;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                res = Math.max(res, getMax(i, j, arrays, m, n));
            }
        }
        System.out.println(res);
    }
}
```

```
    static int getMax(int x, int y, List<List<Integer>> list, int m, int n) {
        if (list.get(x).get(y) == 0) {
            return 0;
        }
        int count = list.get(x).get(y);
```

```

list.get(x).set(y, 0);
int[][] flag = {{1, 0}, {0, 1}, {-1, 0}, {0, -1}};
for (int[] f : flag) {
    int fx = f[0];
    int fy = f[1];
    int xx = fx + x;
    int yy = fy + y;
    if (0 <= xx && xx < n && 0 <= yy && yy < m) {
        count += getMax(xx, yy, list, m, n);
    }
}
return count;
}
}

```