

套路大解密：搞懂题目考察什么，剩下的任务就是套模板！

[负雪明烛](https://leetcode-cn.com/u/fuxuemingzhu/)发布于 5 小时前531BFSC++

如果你经常在想为什么别人做题那么快，其实答案很简单：一是要快速明白过来这个题在考察什么，这需要对常用的数据结构和算法的适用场景很熟悉；二是要背模板！没错，大神也需要背模板！当然，是理解着背，不是死记硬背！

我在[我的博客](https://blog.csdn.net/fuxuemingzhu" \t "_blank)中经常说，弄懂题目想考察什么就解决了一半问题，剩下的一半问题就是套模板。

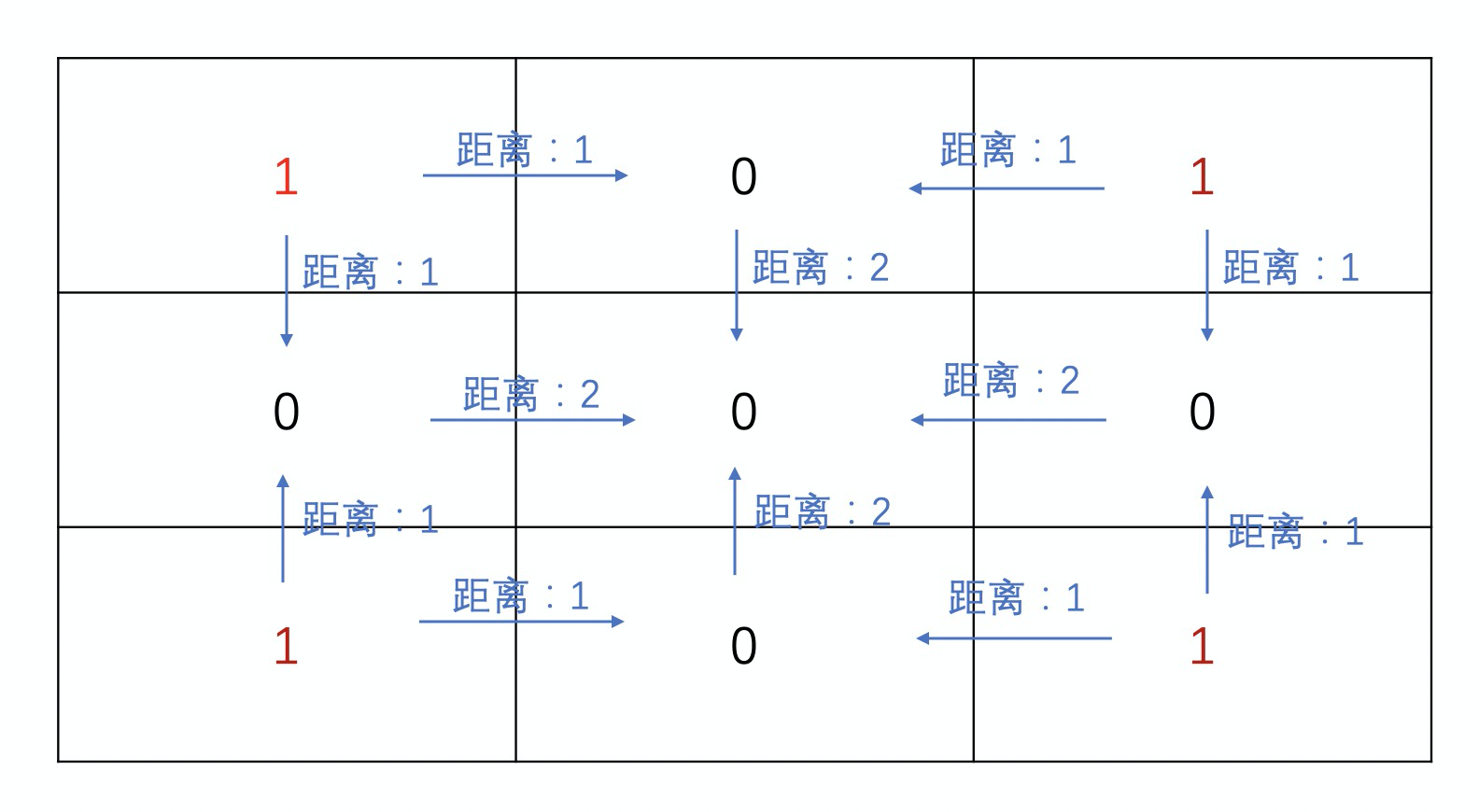
下面，我就按照这个思路给大家展示，如何快速解决这个题。

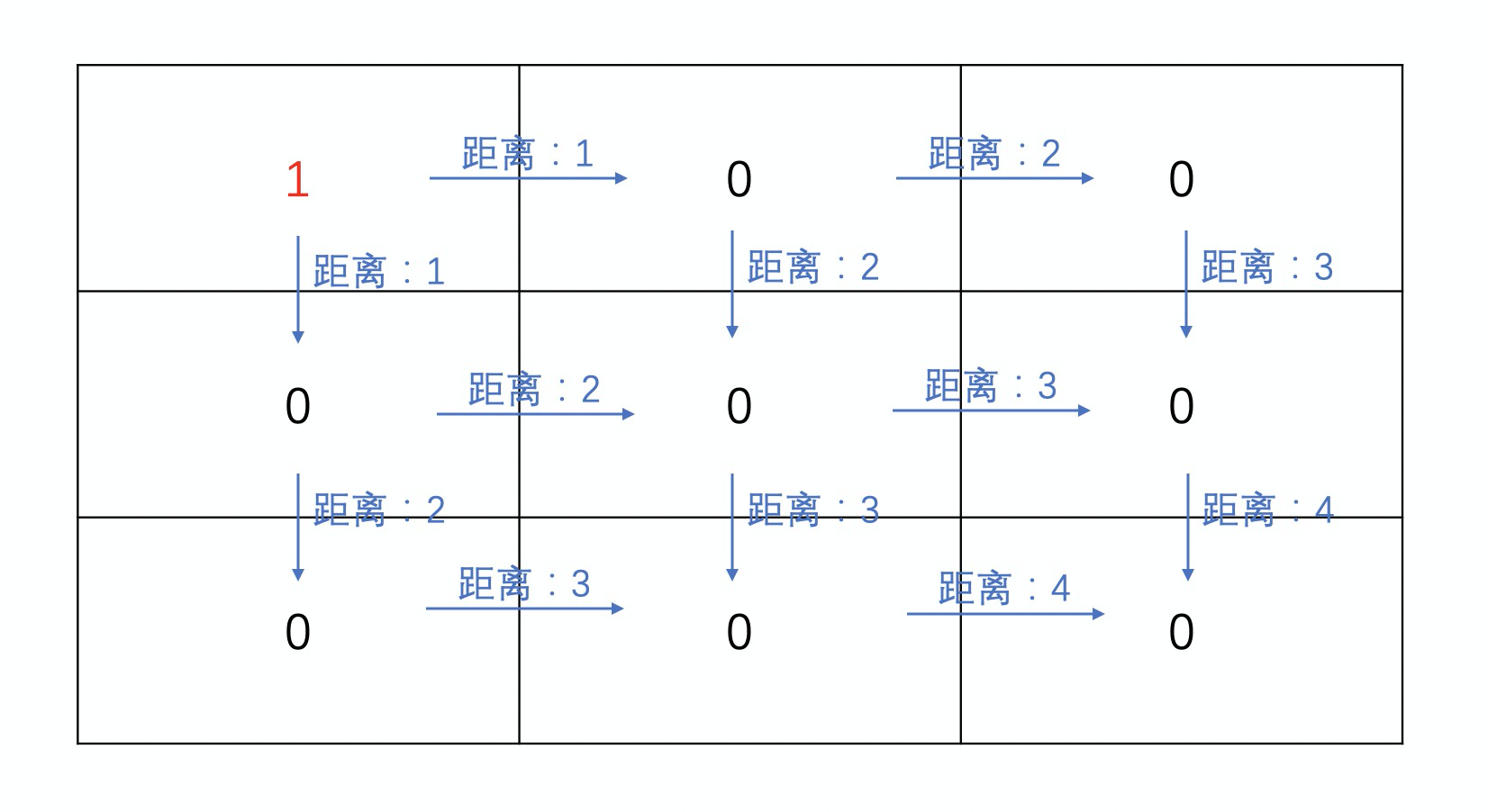
**这个题想考察什么？**

虽然题目千变万化，但是考察点永远是那几个。本题给出了一个场景：求所有海洋点到离它最近的陆地点的距离的最大值。那么我们求出每一个海洋点到其最近陆地点的最短距离，在这些最短距离中找最大值不就好了么？

在向下阅读之前，一定要确保你理解了题意。其中曼哈顿距离就是只能沿着横、竖到达另外一个点走的步数。

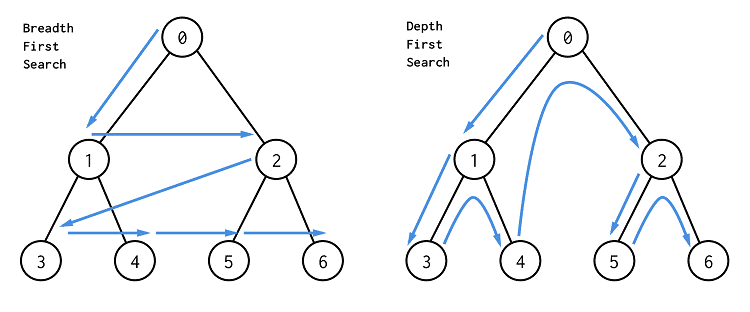
题目给出的两个示例：

example 1：  
  
题目所求是中间那个0，距离所有1的距离最大为2.

example 2：  
  
题目所求是有下角那个0，距离所有1的距离最大为4.

在一个图中，能从一个点出发求这种**最短距离**的方法很容易想到就是BFS，BFS的名称是广度优先遍历，即把周围这一圈搜索完成之后，再搜索下一圈，是慢慢扩大搜索范围的。

图左边是BFS，按照层进行搜索；图右边是DFS，先一路走到底，然后再回头搜索。



题目给出了多个陆地，要找出每个海洋点A到陆地B的最近曼哈顿距离。由于A到B的距离和B到A的距离一样的，所以其实我们可以换个思维：找出每个陆地B到所有海洋点A的距离，对每个海洋点A取最小距离就好了。

因此，题目可以抽象成：多个起始点的BFS。恭喜你已经解决了一半问题。

**剩下的任务就是套模板！**

我在博客中已经总结了所有常见的算法模板，[【LeetCode】代码模板，刷题必会](https://blog.csdn.net/fuxuemingzhu/article/details/101900729" \t "_blank)，直接拿来用！

BFS使用队列，把每个还没有搜索到的点依次放入队列，然后再弹出队列的头部元素当做当前遍历点。BFS总共有两个模板：

1. 如果不需要确定当前遍历到了哪一层，BFS模板如下。

while queue 不空：

cur = queue.pop()

for 节点 in cur的所有相邻节点：

if 该节点有效且未访问过：

queue.push(该节点)

1. 如果要确定当前遍历到了哪一层，BFS模板如下。  
   这里增加了level表示当前遍历到二叉树中的哪一层了，也可以理解为在一个图中，现在已经走了多少步了。size表示在当前遍历层有多少个元素，也就是队列中的元素数，我们把这些元素一次性遍历完，即把当前层的所有元素都向外走了一步。

level = 0

while queue 不空：

size = queue.size()

while (size --) {

cur = queue.pop()

for 节点 in cur的所有相邻节点：

if 该节点有效且未被访问过：

queue.push(该节点)

}

level ++;

上面两个是通用模板，在任何题目中都可以用，是要记住的！

上面说了这个题是多个起始点的BFS，不要害怕，就是需要先遍历一遍矩阵，把所有陆地先放进队列中，然后再利用模板二。

至此，把上面的思路套进模板，题目就能解决了。

在C++中可以使用queue作为队列。我下面使用的是deque双端队列，但是只当做单端的队列来用。

C++代码如下，如果看不懂C++也不要紧，注释很详细。

class Solution {

public:

int maxDistance(vector<vector<int>>& grid) {

const int M = grid.size();

const int N = grid[0].size();

// 使用deque作为队列

deque<pair<int, int>> deq;

for (int i = 0; i < M; ++i) {

for (int j = 0; j < N; ++j) {

if (grid[i][j] == 1) {

// 将所有陆地都放入队列中

deq.push\_back({i, j});

}

}

}

// 如果没有陆地或者海洋，返回-1

if (deq.size() == 0 || deq.size() == M \* N) {

return -1;

}

// 由于BFS的第一层遍历是从陆地开始，因此遍历完第一层之后distance应该是0

int distance = -1;

// 对队列的元素进行遍历

while (deq.size() != 0) {

// 新遍历了一层

distance ++;

// 当前层的元素有多少，在该轮中一次性遍历完当前层

int size = deq.size();

while (size --) {

// BFS遍历的当前元素永远是队列的开头元素

auto cur = deq.front(); deq.pop\_front();

// 对当前元素的各个方向进行搜索

for (auto& d : directions) {

int x = cur.first + d[0];

int y = cur.second + d[1];

// 如果搜索到的新坐标超出范围/陆地/已经遍历过，则不搜索了

if (x < 0 || x >= M || y < 0 || y >= N ||

grid[x][y] != 0) {

continue;

}

// 把grid中搜索过的元素设置为2

grid[x][y] = 2;

// 放入队列中

deq.push\_back({x, y});

}

}

}

// 最终走了多少层才把海洋遍历完

return distance;

}

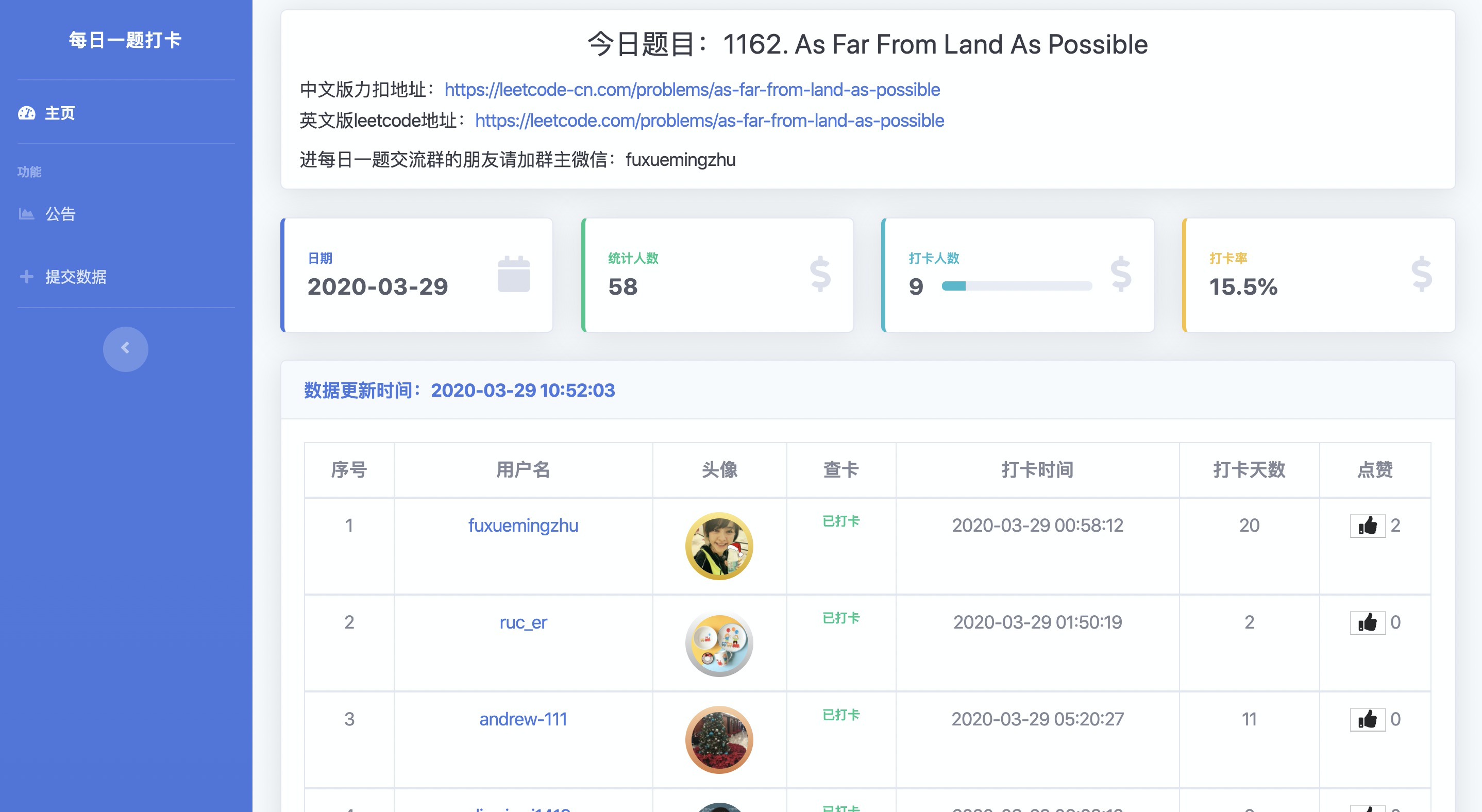
private:

vector<vector<int>> directions = {{-1, 0}, {1, 0}, {0, 1}, {0, -1}};

};

欢迎关注[负雪明烛的刷题博客](https://blog.csdn.net/fuxuemingzhu)，刷题800多，每道都记录了写法！

力扣每日一题活动建群啦，一起监督和讨论，我自建监督网址：<http://140.143.79.116/>，加入方式可以在监督网址中看到。



下一篇：理清思路：为什么用 BFS、如何写 BFS 代码求最短路径（Java/Python）

© 著作权归作者所有

6

条评论

最热



编辑

预览

评论

[[](https://leetcode-cn.com/u/bu-hui-er-cha-shu/)不会二叉树](https://leetcode-cn.com/u/bu-hui-er-cha-shu/)

2 小时前

厉害，老哥，你的链接已收藏

赞踩查看 1 条回复回复

[[](https://leetcode-cn.com/u/jonahmoon/)jonahmoon](https://leetcode-cn.com/u/jonahmoon/)

（编辑过）4 小时前

请教一下，题目中问的是(1)某块海洋到陆地的距离最远，然后求其到陆地的最近距离； 题解中直接把这个最远距离定义成(2)只要某块海洋到陆地的最小距离最大，则其到陆地距离最远； (1)和(2)是怎么等价的，怎么证明呢？这个到陆地距离最远到底是怎么定义的，平均距离最大且方差最小吗？ 举个例子： 101 000 0A0 000 010 和 101 000 000 0B0 010 中A和B哪个到陆地的举例最远呢？

赞踩查看 1 条回复回复分享举报