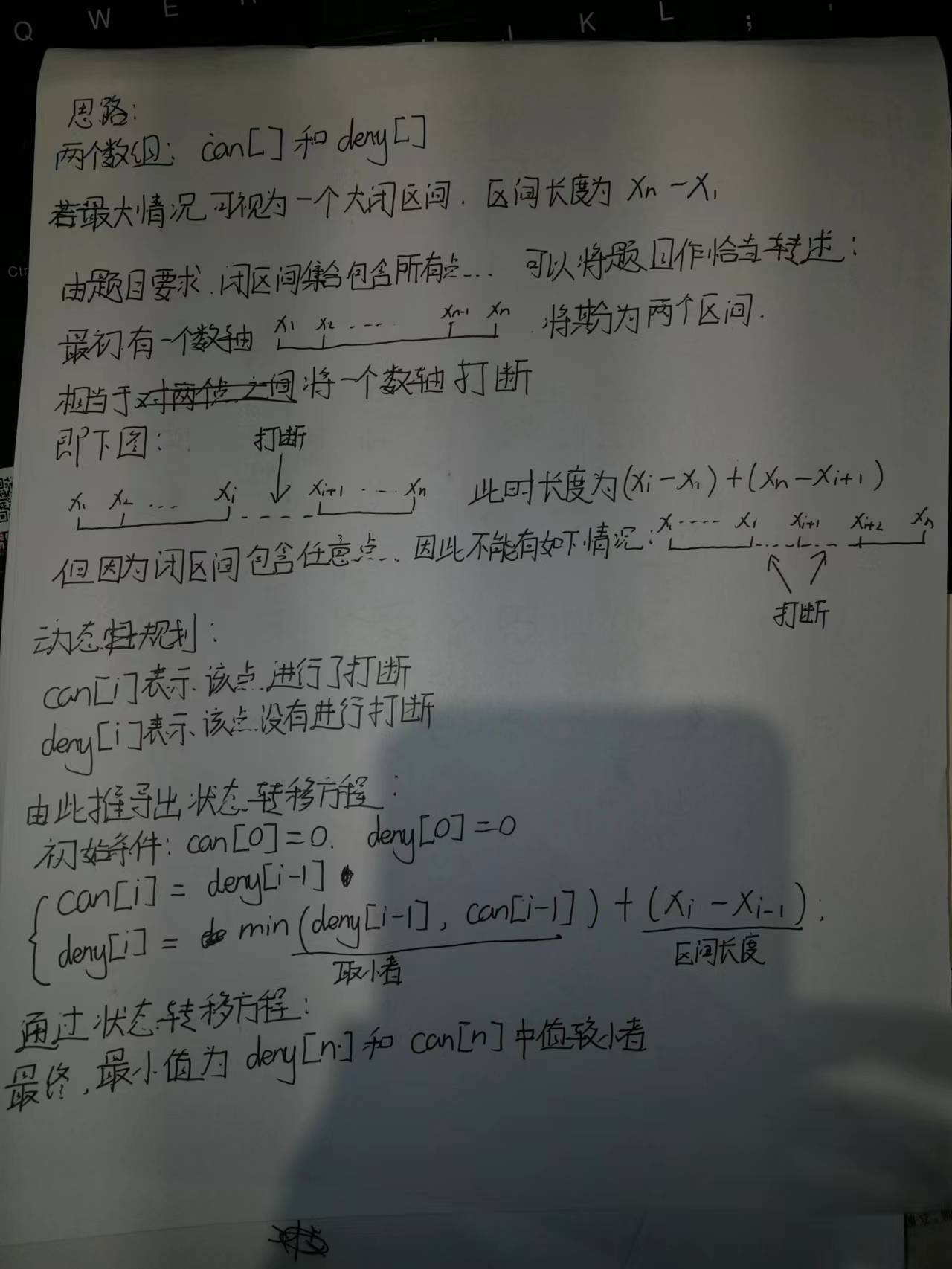
第二次作业

1. 设计一个高效算法，对实数线上给定的一个点集，求一个单位长度闭合区间的集合，包含所有给定的点，并要求此集合最小，证明你的算法的正确定。

**算法思路：**



**正确性：**

**区间的打断只有一个约束条件：一个点的两侧不能同时被打断（否则只剩下一个点，这个点不被任何闭区间）**

**因此由约束条件可知，一个区间能不能被打断，唯一的约束条件是与该区间相邻的区间不能被打断**

**因此使用状态转移方程[i]保存的最小值，就是从0到i点的最小值**

**效率分析：显然，状态转移方程下效率为O（n）**

1. 给定一个非负整数数组，你最初位于数组的第一个位置。数组中的每个元素代表你在该位置可以跳跃的最大长度。判断你是否能够到达最后一个位置。写出算法思路，并给出伪代码。

示例： 【2**, 3, 1, 1, 4】可达**

【3**, 2, 1, 0, 4**】不可达

解：

与leetcode.403:青蛙过河（https://leetcode.cn/problems/frog-jump/）异曲同工

**算法思想**：最初的时候从第一个位置出发，进行深度优先递归操作，在每个点都访问一切可访问的点；在DFS操作结束后，只需检查最后一个位置是否被访问过即可判断能否达到；

**效率**：根据题目，跳跃过程是无记忆的（当前可跳跃距离只由当前位置的可跳跃距离决定，不受上一步操作的影响），所以每个节点只需访问一次，理论效率O（n）;

-------------------------------------------------------------------------------------------

Find(0);//初始点开始DFS

Return Point[Point.length-1].hasReach;//最后一个点是否被访问

find(int index):

if(Point.index.hasReach):

return;//已经访问过

else:

Point[index].hasReach = true;//访问标记

width = Point[index].length;//可跳跃距离

for i = 1 to width:

find(i+index);//继续往下dfs

end;

end;

-------------------------------------------------------------------------------------------