

Audience Question

Q: 看不到屏幕。。。

A: 现在能看到了嘛

Audience Question

Q: 这道题用并查集做的，我 debug 了一天最后没办法找问答老师帮我解决了一个 bug

A: 好像是我回答的，嘿嘿

Audience Question

Q: Nearest Exit 这题官网答案还没更新

A: 这题最近就会更新，稍等一下

Audience Question

Q: 老师有声音吗？

A: 有声音呀

Audience Question

Q: 很清楚

A: 赞

Audience Question

Q: A

A: 正确答案

Audience Question

Q: 小->大

A: 对的

Audience Question

Q: binary search

A: 正确答案

Audience Question

Q: 这题不是讲了么

A: 啥时候讲的?

Audience Question

Q: 这题放到了第三节的 ladder 里了, 是不是被误以为讲过了

A: 哈哈, 那位同学肯定是提前预习过了

Audience Question

Q: 所以这个题目和 island 的区别就是是否考虑包围?

A: 差不多

Audience Question

Q: 虽然当做模板背下来了, 但为什么 $mid = l + (r-l)/2$ 能保证不溢出呢...?

A: $(l+r)/2$ 中 $l+r$ 在 l, r 很大时有可能溢出, $l+(r-l)/2$ 不会, 中间结果不会大于 r

Audience Question

Q: 2 没有连着, 到不了吧

A: 2 是第二个开始注水的位置

Audience Question

Q: union find 和 BFS 对于这类题目是不是还是 union find 比较快呢?

A: 不是所有 union find 都适合这类题目

Audience Question

Q: 只要从最外圈的 0 进行 BFS 就行了吧??到不了的内圈 0 就变成 X

A: 思路正确

Audience Question

Q: how do u decide 1 and 2

A: 两个区域不连通

Audience Question

Q: $n * m + 1$ 为什么+1

A: 多一个元素肯定不会错嘛

Audience Question

Q: surrounded 是 i 一维 boolean 数组

A: 因为 surrounded 代表这个下标为 i 的区域是否被 X 包围

Audience Question

Q: No 变量是不是没有必要，感觉 01 就行了

A: 不行呀，No 是当前的联通块的标号

Audience Question

Q: 会不会 stack overflow?

A: 不会的

Audience Question

Q: 两个 queue 不如定义一个 node 好看。。

A: 好建议!

Audience Question

Q: 可不可以从边界所有的 O 出发 bfs，把与边界 O 链接的 O 全部 mark，然后最后没 mark 的 O 就是要填充的

A: 也是可以的，只不过需要的空间多一点，操作复杂一点

Audience Question

Q: 因为 surrounded 代表这个下标为 i 的区域是否被 X 包围--- board 是 2 维的，不是 1 维的。你如何用一位的 surrounded 来标记？

A: surrounded 代表这个下标为 i 的联通块是否被 X 包围，就像刚刚演示的 1 和 2

Audience Question

Q: No 是什么

A: urrounded 代表这个下标为 i 的联通块是否被 X 包围

Audience Question

Q: 往右和往下两个方向扩展不行么

A: 结果是一样的，不过联通块的个数就不对了

Audience Question

Q: 就是每个点只访问一次？

A: 是的，这是 bfs 的精髓

Audience Question

Q: 可以从四边开始注水，被水淹的地方暂标成 F，完成后扫一遍全图，如果还是 O 的，就把它标为 X；如果是 F，把它重新标为 O，这样思路再简单些。

A: 思路是对的，但是要重新处理整个图

Audience Question

Q: 为什么要用 No

A: 方便 bfs 完回来把所有被包围的联通块标记为 X 用的

Audience Question

Q: 这样的题是 BFS 好还是 DFS

A: BFS 好

Audience Question

Q: surrounded 表示的是, bfs 之后, 标记为 1 / 2/3/4。。。的区域, 是不是被包围。有可能不被包围, 就是 false, 是吧。就不改成 X。懂了大概

A: 很正确!

Audience Question

Q: running time 是什么?

A: 运行时

Audience Question

Q: 不不不, 我是说这道题的 running time

A: 时间复杂度吗? $O(nm)$

Audience Question

Q: 为什么从边界出发的空间复杂度会搞?

A: 我看懂了, 你说的方法应该没问题, 复杂度和老师讲的一样

Audience Question

Q: 反向考虑指的是用没有和边界连接来确定是 surrounded 吗?

A: 找到和边界连接的联通块的标号, 把除了这些标号的联通块标记为 X

Audience Question

Q: 能不能不要只讲 java?

A: solution 里有 C++ 的答案哦

Audience Question

Q: 时间复杂度不是 running time，是 time complexity :)

A: 我的锅。。可以把答案跑一下看看

Audience Question

Q: 可不可以把房间当成“注水点”，一层层去找门

A: 可以的

Audience Question

Q: no sound?

A: 有啊

Audience Question

Q: 这个需要标记 mark 吗？就是这个点访问过了

A: 访问过了会有一个当前的最短距离，相当于 mark

Audience Question

Q: Queue<int[]> q = new LinkedList();

A: 没见过这样的写法。。

Audience Question

Q: 超级源点在代码怎么体现

A: 开始的时候把所有门的位置入队列

Audience Question

Q: 没有声音了

A: 有声音哦

Audience Question

Q: dummy 点边的长度应该是 0 吧?不然距离不是多算了

A: 是的, 这是个虚拟的点

Audience Question

Q: 老师的代码是不是已经在九章网站的答案里贴出来了?

A: 是的

Audience Question

Q: 不需要比较一下吗, 有些点是从任何源点都能到达的, 不需要比较一下, 找出最短路径吗

A: 这个点在第一次访问到的时候就是最短的路径

Audience Question

Q: surrounded region 那道题 如果用 并查集 有什么优势吗?

A: 不如 bfs 呀

Audience Question

Q: 什么佛洛依德?

A: 可以搜索一下 floyd 最短路

Audience Question

Q: what is bingchaji in english?

A: union find

Audience Question

Q: 如果边长不是 1 怎么办?

A: 用其他最短路算法做

Audience Question

Q: 第一题如果也用超级源的话会变快吗?

A: 第一题没必要呀, 单纯的 bfs 就行了

Audience Question

Q: 是类似 dijkstra algorithm 吗

A: 对的

Audience Question

Q: 那做题目的时候有什么 preference 吗? 在 union find 和 BFS 之间

A: union find 最好是解普通图上的联通性问题

Audience Question

Q: 静态图, 单词查询用 bfs, 动态图, 多次查询用并查集

A: 赞

Audience Question

Q: 老师能提一提边长不为 1 的时候求最短路径的方法么

A: 可以用其他的求最短路的方法, 如 dijkstra, spfa, floyd

Audience Question

Q: 边长有权重的话是不是就是路由 routing 用的算法

A: 差不多

Audience Question

Q: 啥时候用 union find 去 check 连通性?

A: 点不是用矩阵的方式给出的，比如树形图

Audience Question

Q: why i cannot find the problem and solution for problem "nearest exit" ?

A: 可能这题答案还没加上去，稍等一会，我们之后会加上

Audience Question

Q: 静态图和动态图是什么？可以举例吗？

A: 静态图大概是在数据中给定的，之后不会改变的图，反之会改变的是动态图

Audience Question

Q: 为什么要减 '0' 呀？

A: 把 ascii 码转换为数字

Audience Question

Q: 我是用 c++ 的，Java 不需要用引用吗？

A: java 除了基本类型是传值，其他都是引用

Audience Question

Q: 退出条件也可以比较 str 的长度与号码长度是不是相等吧

A: 也是可以的

Audience Question

Q: 当白板代码的时候，先写拓展或先写 return，要留多大空间在前面是个问题诶。。

A: 这个要自己把握

Audience Question

Q: 成员变量的话如果 `test case` 是反复 `call` 我们的 `method` 就会错吧

A: 每次使用之前都初始化就行了

Audience Question

Q: `phone` 那个数组里怎么两个空字符串

A: 因为数字 0 和 1 没有对应的字符

Audience Question

Q: `str` 为什么不用 `remove` 掉 `character` 呢

A: 因为 `str` 是参数，不需要回溯

Audience Question

Q: `str` 为什么不用 `remove` 掉 `character` 呢？答案：`str` 是 `immutable`

A: 对的

Audience Question

Q: 当前层次 `x` 是不是可以省略，根据 `str` 的长度可以推出 `x`

A: 这样写比较清晰

Audience Question

Q: 面试中使用全局变量会不会让人觉得你的代码风格不好？之前好像有听过这种说法

A: 的确是这样的

Audience Question

Q: 所有 `backtracking` 的题都能用 `dfs` 么

A: 应该都是可以的

Audience Question

Q: 这个题目在枚举过程中是不是也要试一下 3

A: 的确试了 3 呀

Audience Question

Q: 回溯 英文是啥

A: backtracking

Audience Question

Q: 为什么前面一题不需要回溯?

A: 因为字符串变量定义以后就不能改变, 下一层 dfs 的时候是传一个新的 str 进去, 旧的那个还在原来这层的 dfs 里

Audience Question

Q: 这道题 n 可以是负数吗

A: 不会是负数, 不然答案里的 factors 也要是负数了

Audience Question

Q: code 里面, 貌似没有把 2 加入 ans——item 的地方啊

A: 当 i=2 的时候 ans_item.add(i)

Audience Question

Q: 这道题 如果 输入的数是负数呢?

A: 这题不考虑负数

Audience Question

Q: 老师, 那为什么上一题的 str 可以改变呢

A: str 不能改变呀 str1+str2 是构造了一个新的 str

Audience Question

Q: 退出条件能不能再解释下

A: 剩下的 product 不能被整除的时候

Audience Question

Q: 在那个 for 里面，是不是， when lastF = 2 传入的

A: 是的

Audience Question

Q: 新的 new_ans_item 要删除掉吧？否者有内存泄露？

A: 会自动释放的，这个不用担心

Audience Question

Q: ans 也可以放在 getF 里面啊。。。

A: 不用全局也可以啊，只要放在参数里就行

Audience Question

Q: Input: 1073741824

A: 这个数据咋了

Audience Question

Q: 参数里面放 ans_item 然后不 new list 还和之前一样去回滚会有什么问题吗？没明白为什么要 new list

A: 这个是深复制 deep copy

Audience Question

Q: 为什么说 23，写成 24，老师要注意身体啊。是征兆。

A: 很幽默

Audience Question

Q: 不好意思没说清楚，就是把 `ans_item` 放进 `function parameter` 以后，可以继续用 `add`，`dfs (ans_item), remove` 这样，而不用重新 `new` 一个 `list` 么？

A: 不 `deep copy` 的话之后改变这个 `list`，`ans` 里的都会改变

Audience Question

Q: `lastFactor` 的定义是什么啊？

A: 顾名思义，最后一个因子

Audience Question

Q: 可不可以再说一下为什么 `=='0'` 的情况最后写？

A: 假如字符串只有一个 `0`，先判断 `0` 就错了

Audience Question

Q: 所以数字个数的区别比如 5 个数字 `12345`，是 `1, 2345`；`1, 2, 345`；`1, 2, 3, 4, 5`，这样 `loop` 的顺序吗？

A: 不是，是枚举所有的拆分方法

Audience Question

Q: 这题好难

A: 的确是 `hard` 的难度

Audience Question

Q: 在强化班里这道题不是还要现建立个 `Trie` 树。好麻烦的

A: 多写写就不麻烦了，哈哈

Audience Question

Q: 这题不是要用 `trietree` 么

A: 可以用 `trie`

Audience Question

Q: 剪枝可以理解为 DP 吗?

A: 这是两个概念呀

Audience Question

Q: 这道题要 prefix search 是不是用 Trie 比较好

A: hash 和 trie 都可以的

Audience Question

Q: trie 树是什么

A: trie, 又称前缀树或字典树, 是一种有序树, 用于保存关联数组, 其中的键通常是字符串。

Audience Question

Q: DFS 是不是最难的 onsite 题目, 没有之一?

A: 这个不一定呀

Audience Question

Q: 处理 prefix 那么如何得到 le la 结尾的单词?

A: 继续 dfs

Audience Question

Q: 滚动和回溯什么关系?

A: 应该是同一个概念说法不同

Audience Question

Q: 很多题放到参数中一样要滚动啊

A: 具体题目具体分析呗

Audience Question

Q: Nearest-exit 答案看不了

A: 我们之后会上传，稍等一下

Audience Question

Q: 每层占用空间大的时候不应该每一层复制吧，不然空间用得太多

A: 对的，对象比较大的情况就要传引用

Audience Question

Q: 不滚动的情况，是不是传递了参数本身，每一层都复制了一遍

A: 是的

Audience Question

Q: 但是如果 `string` 很长的话，每层都要复制一次，这样会增加时间复杂度，不应该用 `stringbuilder` 来做更好一些？

A: 也是可以的

Audience Question

Q: 在 `java` 里，`dfs` 传参时，`primitive type` 比如 `int`, `String` 都是传参数本身，`object` 都是传 `reference`，对吧？

A: `String` 也是传引用

Audience Question

Q: 会爆栈吗？

A: 不会的哦

Audience Question

Q: 谢谢老师助教

A: 不用谢~

Audience Question

Q: 引用，c++的么？

A: c++和 java 都有