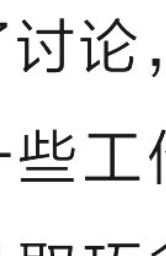


第173封信 | Google的面试题：推棋问题

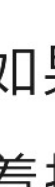


吴军



第173封信 | Google的面试题..

10:53 5.10MB



信件朗读者：宝木

小师弟，你好！

在过去的三天我们围绕迷宫问题展开了讨论，通过迷宫问题，我强调一点：如果一些工作我们无法省略，最好不要想着投机取巧省掉它们，因为那样就得不到我们想要的东西了。工作中最可怕的不是进度慢，而是返工。

今天我们利用Google的一道面试题，将这个做事方法再进行一次讨论。这道题我昨天已经提前给大家了，如果你听完之后没有做出来，不用担心，因为我在Google用它面试10个人，也就一小半做得出来，而我所面试的人已经是HR筛选过，并且经过电话面试的。

Google的这道面试题是这么出的：

在一个有起始点、没有终点的棋盘上，放着一列列数量不等的围棋子。在下图中，前五列分别有3、2、1、7、4、1、1、4、3个棋子。注意，**这个棋盘有下面的边界，左边的边界，但是没有上面的边界和右边的边界**，虽然我用了围棋盘，往上和往右都只有19根线，但这个题目并没有这样的约定，往上和往右可以无穷大。

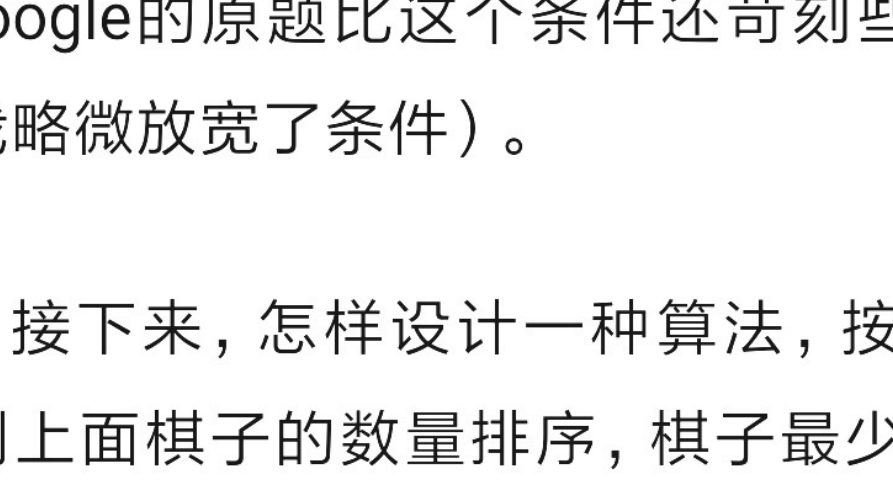


图1

现在有一个小机器人，它可以沿着横线和竖线上下左右走动，在有棋子的交叉点捡起一个棋子，放到自己的口袋中，或者在没有棋子的地方放下一个棋子，当然后一个动作的前提是他的口袋里有棋子（Google的原题比这个条件还苛刻些，这里我略微放宽了条件）。

接下来，怎样设计一种算法，按照每一列上面棋子的数量排序，棋子最少的一列在左边，最多的一列在右边。注意：对于右边那些没有占用的纵列，是不应该放棋子的。也就是说，如果一开始最右边的棋子在从左边数第N列，排完序之后，最右边的依然应该在那一列，从第N+1列开始都应该是空白，就如下图所示。

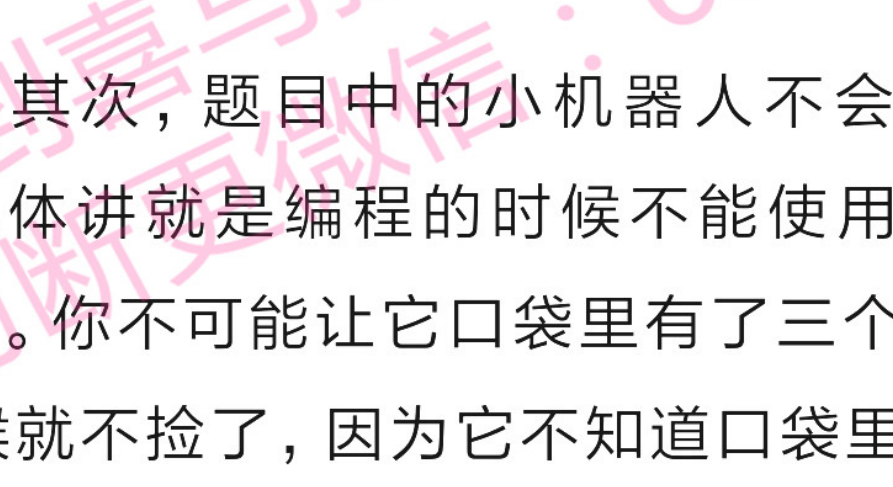


图2

我们以前讲过，面试的时候首先要把问题搞清楚，因此我们先来理解一下这个问题。

首先，这个问题看似是排序问题，但是，我们过去讲排序时候说的比较大小的功能它都没有。因此不能直接用排序算法实现。

其次，题目中的小机器人不会数数（具体讲就是编程的时候不能使用计数器）。你不可能让它口袋里有了三个球的时候就不捡了，因为它不知道口袋里有多少球。也不可能往前走两步就停下来，因为它数不清楚步数，它要么往某个方向走一步，要么一口气走下去，直到某个条件满足。从题目中你可以看出，它只有非常有限的的能力，而你需要在它的能力范围内把这件事情完成了。

题目的要求清楚了，我们节省时间，直接讲解答案了。

如果对比上面的两张图，你其实会发现第二张图中的棋子的位置，就是将第一张图中所有棋子按横向推到最右边。那么这个排序的问题，就变成了移动棋子和判断边界的问题。

我们一排一排地从上往下看，最上面一排，也就是从底下数第七排，有一个棋子在第四列，我们需要移到第九列。如果这个干活儿的小机器人可以数数，当然这件事很容易做，遗憾的是它不识数，这是我们要解决的第一个难题。

等我们往下动第四排时，情况就又有有点复杂了。虽然我说“把棋子往右边推”，但是在计算机中，并没有一个“推”的装置，所谓的“推”，就是要将第四、第五列的棋子捡起来，填到右边的两个空位中。由于无法数数，不知道要捡起几个棋子，也不知道，放到哪个位置。这是第二个挑战。

你看一下下面这张图，就容易理解上述两种不同情况的操作了，黑色空心的圆圈代表原来的棋子位置，红色实心的代表新的位置。

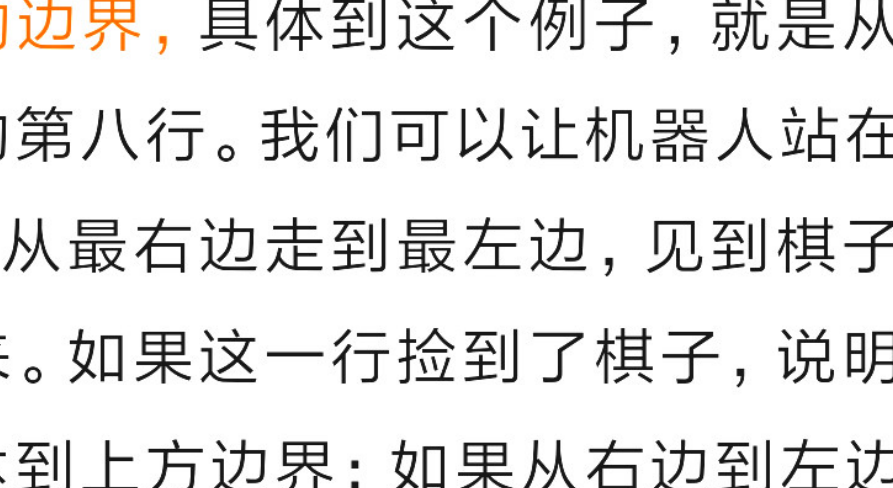


图3

我在面试Google的求职者时，大部分人能想到这一步。但是，绝大多数人试图将第三张图中那些在空心圆圈的棋子一个个地搬运到实心红圆圈去，但是这其实在没有计数功能的情况下做不到。正确的做法应该是这样的，我们分两步说明，先说明判定边界的原理。

首先，**需要知道如何判定将来棋子右边的边界**，具体到这个例子，就是从左数第十列。由于每一列至少有一个棋子，因此如果沿着最下面的一排从左往右走，看到前面空的位置，就知道到了右边的边界了。但是，对于第二排、第三排.....在没有排好次序之前，不能作这样的假设。也就是说，要判断最右边的边界，每次都需要走到最下面，从左到右走一遍。这当然是个笨办法，但事实上并不更坏。

其次，**需要知道如何判断将来棋子上方的边界**，具体到这个例子，就是从下往上的第八行。我们可以让机器人站在某一行，从最右边走到最左边，见到棋子就捡起来。如果这一行捡到了棋子，说明还没有达到上方边界；如果从右边到左边走过去，一个棋子都没有捡到，说明到了上方的边界。在这个例子中，如果机器人站在第六行，从右走到左，可以捡到棋子，说明还没有遇到上方边界，但是如果站在第八行，一行走下来捡不到棋子，说明遇到了上方边界。这也是一个笨办法，但是其实也不坏。

最后，需要说明由于机器人不会计数，因此，它不能记住右边界在第10列，上方边界在第8行，它只能在到达边界时，用上述方法判断。

有了这三个铺垫，就可以开始移动棋子了。接下来是算法的步骤：

第一步，来到右边界：从最下面一排开始，从左走到右，一直走到右边界停下，即停在第九列。然后往上走一步，来到下一个要整理的那一排。在初始的时候，最底下一排是天然整理好的，不用操心，因此往上走一步就到了第二排，并且是第二排的最右边。

第二步，捡棋子：在当前的一排，从右到左走，如果见到棋子，就捡起来装到自己的口袋里，走到最左边时，这一排所有的棋子能捡的自然都捡光了。如果一个棋子都没有捡到，说明任务完成，程序结束。如果捡到了棋子，这时，这一排所有的棋子都装在了口袋里，位置都是空的。

第三步，将口袋里所有的棋子放到这一排的右半边，一个位置一个。

这一步的难点是，由于这一排棋子都空了，我们不知道右边的边界在哪里。所幸的是，下面一排右边的边界是知道的，因为从左往右，从有棋子到没有棋子的地方，就是边界。

因此为了在当前的一行把棋子放在合适的位置，我们先要往下走一行，走到最右边，再往上走，回到当前的这一行，这兜了一个大圈子，其实就是为了判断右边的边界。下图中，红箭头表示在捡完第三行的棋子后，绕一圈回到这一行最右边的方法。