鸡蛋掉落问题

原创 徐凯悦 LOA算法学习笔记 2021-02-23 19:16

鸡蛋掉落问题是谷歌的一道经典面试题，该问题在leetcode中的描述为：

你将获得K个鸡蛋，并可以使用一栋从1到N共有N层楼的建筑。

每个蛋的功能都是一样的，如果一个蛋碎了，你就不能再把它掉下去。

你知道存在楼层F，满足0≤F≤N任何从高于F的楼层落下的鸡蛋都会碎，从F楼层或比它低的楼层落下的鸡蛋都不会破。

每次移动，你可以取一个鸡蛋（如果你有完整的鸡蛋）并把它从任一楼层X扔下（满足）（满足1≤X≤N）。

你的目标是确切的知道F的值是多少。

无论F的初始值如何，你确定F的值的最小移动次数是多少。

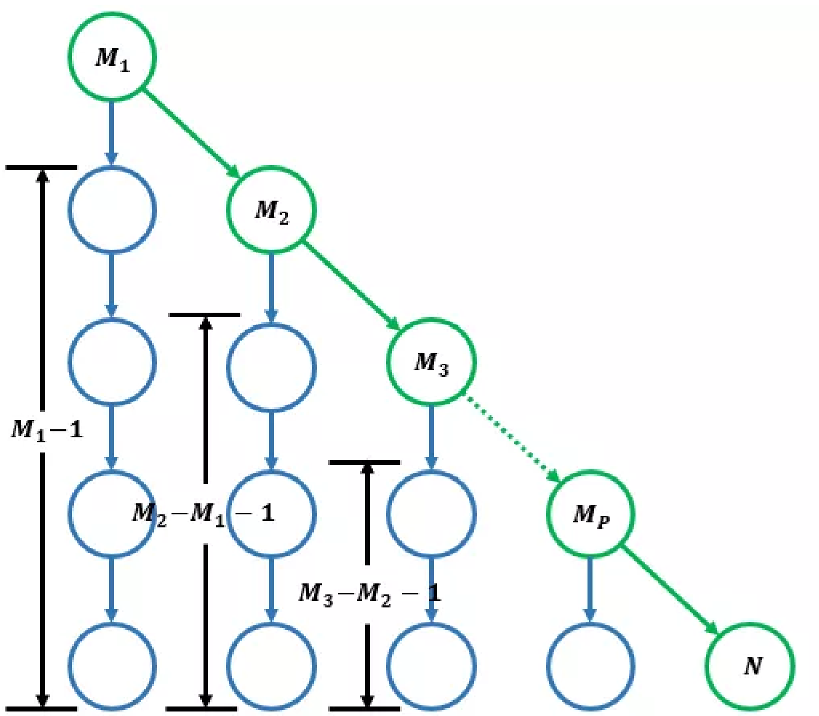
分析题意可知，该问题的目标为在最坏情况下最小化鸡蛋下落次数。一种想当然的方法是利用二分法求解，但进一步分析可知，当$K<\lceil\log_2(N+1)\rceil$时，在最坏情况下会出现所有鸡蛋均已打碎但未找到F的情况，因此该方法不可行。

1. 在求解此问题前，先考虑该问题的一个简单形式：假设只有一个鸡蛋。

这种情况下，只能从第一层开始依次尝试，直到找到临界层。最坏情况下的鸡蛋最小下落次数为N。

2. 其次，考虑有两个鸡蛋的情况。

在这种情况下，可以先利用其中一个鸡蛋进行"试错"，并利用另外一个鸡蛋按照只有一个鸡蛋的方法求解鸡蛋最小下落次数。假设第一个鸡蛋在第M_1层碎了，则将第二个鸡蛋遍历的范围缩小到$[1,M_1)$这一较小的范围。同理，假设第一个鸡蛋在第层没有碎，而在第层碎了，则将第二个鸡蛋遍历的范围缩小到(M_1,M_2)这一较小的范围。以此类推。该过程可用下图描述：



其中绿色轨迹为第一个鸡蛋进行试错时楼层的取值，蓝色轨迹为第二个鸡蛋进行遍历时楼层的取值。可以看出，该决策过程可视为一棵树，而最坏情况下最小化鸡蛋下落次数即为最小化该树的高度。不难证明，要使得当这棵树高度最小，$M_1\sim M_p$中每个节点的两棵子树高度应尽可能接近，即这棵树要尽量满（通过反证法即可证明，不再详细说明）。

当该树满足$M_1\sim M_p$中每个节点的两棵子树高度相等时，该树能判别的楼层数最高为：

$$M_1+M_1-1+M_1-2+\cdots+1=\frac{M_1(M_1-1)}{2}\leq N$$

求解该方程得到的M_1向上取整，即为最坏情况下的鸡蛋最小下落次数。

3. 最后，考虑K个鸡蛋的情况。

通过对有两个鸡蛋的情况进行分析，可知鸡蛋最小下落次数只和楼层高度有关。同理，先利用其中一个鸡蛋进行“试错”，此时该问题为一道动态规划问题。

定义dp[h][k]表示楼高为h，鸡蛋数量为k时最坏情况下的鸡蛋最小下落次数。

在第m（1≤m≤h）层扔下一个鸡蛋，若该鸡蛋碎了，则不需要考虑m+1~h层的情况，该问题转化为楼高为m-1，鸡蛋数量为k-1时最坏情况下的鸡蛋最小下落次数+1，即dp[h][k]=dp[m-1][k-1]+1；若该鸡蛋没碎，则不需要考虑1~m层的情况，该问题转化为楼高为h-m，鸡蛋数量为k时最坏情况下的鸡蛋最小下落次数+1，即dp[h][k]=dp[h-m][k]+1。

因此该问题可利用如下关系式表示：

dp[h][k]=min{max{dp[m-1][k-1],dp[h-m][k]}+1} 1≤m≤h

边界条件：

dp[h][1]=h，即当剩余鸡蛋数为1时，鸡蛋最小下落次数为楼层高度。

dp[1][k]=1，即当楼层高度为1时，鸡蛋最小下落次数为1。

该问题的目标为：dp[N][K]。

此外，经分析可知，当$K \geq \lceil \log_2(N+1) \rceil$时（加1是因为满二叉树的节点个数为2的幂次减1），利用二分查找即可找到满足条件的F，此时最坏情况下的鸡蛋最小下落次数即为$\lceil \log_2(N+1) \rceil$，可提升求解效率。