例 1 - Hanoi

假设有一个 n 层的汉诺塔在左侧的柱子上，要将其移动到右侧的柱子，需要怎么移动最快?

首先，至少，我们需要将最大的，最底下的第 n 圆盘移动到最右侧的柱子上。

由于圆盘只能放在更大的圆盘上，因此为了将第 n 号圆盘移动到最右侧的柱子上，我们需要：

- 第一步，将前 $n-1$ 个圆盘组成的汉诺塔移动到中间的柱子上
- 第二步，将第 n 个圆盘移动到最右侧柱子上
- 第三步，将前 $n-1$ 个圆盘组成的汉诺塔移动到右侧柱子上。

第二步可以直接输出，接下来需要处理的就是第一步和第三步，注意到第 n 层圆盘的存在完全不会影响前 $n-1$ 层圆盘的移动，那么第一步和第三步本质就是 $n-1$ 层的汉诺塔的问题。

那么这 $n-1$ 层的汉诺塔问题又可以再一次化归成 $n-2$ 层的汉诺塔，化归成 $n-3$ 层的汉诺塔……直到变成最基本的情况：只有一个圆盘，直接移动即可。

由此我们就得到了递归关系和初始状态，就能解决整个问题了。

例 5 - 计算小岛面积

我们可以标记小岛外的所有点为 2，同时海岸线上的点起始标记为 1，这样最后统计图上有多少个 0 就可以了。那么该如何标记小岛外的所有点为 2 呢?

首先可以任意找到一个小岛外的点标记为 2，然后可以从这个点出发观察它周围的四个方向：如果目标点**没有到达过**且**不是海岸线（即不是 1）**，那么显然它是之前未到达过的小岛外的点（因为小岛外的点不可能与小岛上的点直接连接，中间一定隔着海岸线）。重复这个过程，直到所有小岛外的点都到达过即可。

那么如何找一个小岛外的点呢？可以利用我们上学期学过的一个技巧，就是在地图外面加一个圈，比如原始地图是这样的：

```
1 1 1 0
1 0 1 0
1 0 1 0
0 1 0 0
```

显然，想要验证某个 0 到底是在小岛内还是在小岛外是无比困难的。假设地图上的点从 $(1,1)$ 到 (n,n) 我们就可以将地图扩展一圈，变成下面这样：

```
0 0 0 0 0 0
0 1 1 1 0 0
0 1 0 1 0 0
0 1 0 1 0 0
0 0 1 0 0 0
0 0 0 0 0 0
```

这样之后，显然 $(0,0)$ 一定是小岛外的点，从 $(0,0)$ 开始出发，对结果也是没有影响的（这个技巧是非常常用的，在我们的代码实现中并没有四周都扩展，只扩展了左面和上面，当然对这道题的结果没有影响）。