



链滴

2 道很有意思的进制题

作者: [lyn](#)

原文链接: <https://ld246.com/article/1498230374649>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. 一共有10只老鼠，1000只水杯，杯子里都盛着水，只有一个杯中是有毒的液体，这种毒液老鼠喝下去以后，一个星期就会死掉。现在要求使用这10只老鼠，在一个星期之内找出盛着毒液的杯子。

分析一下，这道题目其实是一道和二进制相关的题目。由于只能在一个星期内完成任务，那就说明不进行重复的实验。这样的话，由于10只老鼠在一个星期以后的状态只能是死亡或者存活两种状态，根乘法原理，我们知道，10个单位的两种状态，可能出现的所有可能是 $2^{10} = 1024$ 。足够编码1000杯子了。也就是说，我们现在的任务是把1000个杯子与10只老鼠的死和活的组合——对应起来。如把1到1000表示成二进制，那么答案就很明显了。

将1到1000共一千个数，都转换成二进制编码。对于数字A，如果它的二进制的最低位是1，那么就让1号老鼠喝一点这个杯子里的液体，如果是0就不喝。A的二进制的倒数第二低位如果是1，那么编号为2的老鼠就喝，否则不喝。依次类推。直到最高位如果是1，那么编号为10的老鼠就喝，否则该老鼠不喝。一个星期以后观察结果，如果编号为x的老鼠死掉了，那么就从低向高数x位，在那个位置上标1，他的位置标0。例如，如果只有2号和4号老鼠死掉了，那么记录的结果就是“00 0000 1010”，这是一个二进制数。把它转成十进制是10，这就意味着10号杯子里装的是毒液，其他杯子则是水。

2. 有一架天平，它有20个砝码，这20个砝码的重量分别为1,3,9,27... 3^{19} 。只要被称的物品的重量为位于区间 $[1, (3^{20}-1)/2]$ 的整数，就可以使用这架天平进行称量。假设物品一直放在天平的左边。现在给出每个物品的重量，请打印出称量的方案。输出格式为两组数字，第一组代表天平左边要放的砝码，第二组代表天平右边要放的砝码。这两组数中间用空格隔开。每一组内部的数使用逗号隔开。如天平的某一边不需要放砝码，那就打印empty。

例如，输入是9，输出是empty 9，代表左边不需要额外的砝码，而右边需要放一个重量为9的砝码。输入是4，输出是empty 1,3。

从最简单的情况入手进行分析。比如说1，那右边只要放1就可以。如果是2，我们可以用左边放1，右边放3这样的方案代替。3，右边只要放3。4，右边放1和4。到了5，由于不能再使用3-1去凑一个2了，以唯一的方案是左边放1，3，右边放9。同样，6，也只好用9-3来凑。我们来分析一下这个规律。

4的三进制是11，代表了 $1 * 3^1 + 1$ ，5的三进制是12，代表了 $1 * 3^1 + 2 * 3^0$ ，6的三进制是20，代表了 $2 * 3^1 + 0 * 3^0$ 。由于我们手里的砝码是 $3^0, 3^1, 3^2, \dots$ ，也就是说，如果目标数的各个位置上都是1，那我们直接就能组合出来，而如果第n位上是2，就必须使用 $3^{(n+1)} - 3^n$ 来凑这个数。也就是说，我们要在天平的左右两边各加上一个 3^n ，才能保持平衡。这样就把低一位上的，消除成了高一位上的1。

好了。有了这个算法，我们就可以写出程序了。

```
public int[][] solve(int x) {
    int pl = 0, pr = 0;
    int poise = 1, r;
    final int LEFT = 0, RIGHT = 1;
    int[][] result = new int[2][20];

    while (x > 0) {
        r = x % 3;
        if (r == 2) {
            result[LEFT][pl++] = poise;
            x = (x + 1) / 3;
        }
        else if (r == 1) {
            result[RIGHT][pr++] = poise;
            x = x / 3;
        }
        else
    }
```

```
        x = x / 3;
    poise *= 3;
}
return result;
}
```