

2024 年 11 月思维月赛题解

T1 等值序列

此题为签到题，分别统计 1, 2, 3 出现的次数 x_1, x_2, x_3 ，输出 $n - \max\{x_1, x_2, x_3\}$ 即可。

T2 分薯片

对于所有 n 只海狮，每一轮总共会分配到 $s = \frac{n(n+1)}{2}$ 片薯片，于是我们只需要考虑 $m\%s$ 片薯片的分配结果即可。假设当员工给第 k 只海狮分配完之后，剩余不足 $k+1$ 片薯片，于是员工最终能够得到的薯片数量为 $m\%s - \frac{k(k+1)}{2}$ 。

至于 k 的求解可以用循环以 $O(\sqrt{n})$ 的时间复杂度求出：

```
1 int k = 1;
2 while (k*(k+1) / 2 <= m % s) k++;
3 // 此时 m % s - k*(k-1) / 2 即为答案
```

事实上我们可以利用一元二次方程，以接近 $O(1)$ 的时间复杂度求出 k 的值。对于不等式方程 $\frac{k(k+1)}{2} \leq m\%s$ ，整理后可得 $k^2 + k - 2(m\%s) \leq 0$ ，我们知道这里需要求出满足不等式方程的 k 的最大非负整数解，于是 $k = \lfloor \frac{-1 + \sqrt{1 + 8(m\%s)}}{2} \rfloor$ 。

```
1 int k = (sqrt(1 + 8 * (m%s)) - 1) / 2;
2 // 此时 m % s - k*(k+1) / 2 即为答案
```

T3 十进制的二进制数字

此题如果用枚举法验证求解将会超时，用构造法求解非常简单。具体来说，将最高位设置为 1，然后不断地在后面补 0 和 1，直到超出 n 的范围，时间复杂度为 $\log n$ 。

```
1 void dfs(int k) {
2     if (k > n) return;
3     ++ans; // ans 为全局变量，记录答案
4     dfs(k * 10); // 在后面补 0
5     dfs(k * 10 + 1); // 在后面补 1
6 }
```

当然，我们也可以用递推的方式实现构造：

```
1 int f[100005] = 1, p = 1;
2 for (int i = 1; f[p] <= n; i++) {
3     f[++p] = f[i] * 10;
4     f[++p] = f[i] * 10 + 1;
5 }
6 // 此时若 f[p-1] > n 则 p-2 为答案，否则 p-1 为答案
```

T4 二进制数字

从最低位开始向高位递推，递推过程并不复杂：

- 如果当前位为 0，说明是偶数，则答案加 1，一次除以 2 的操作即可消去当前位；
- 如果当前位为 1，说明是奇数，则答案加 2，一次加 1 操作，一次除以 2 操作，可以消去当前位，需要注意加 1 操作会使得下一位进位，此处需要进行进位操作，避免计算错误，不过并不需要将其处理成数值，在字符上直接加即可，于是下一位有可能为 2；
- 如果当前位为 2，说明此位原本是 1，上一位的进位导致此位为 2，那么此位也应向下一位进位，于是当前位的真正数值应该是 0，所以答案加 1。

```
1  int i = s.size();
2  while (--i) {
3      if (s[i] == '0') {
4          ans++;
5      }
6      else if (s[i] == '1') {
7          ans += 2, s[i-1]++;
8      }
9      else {
10         ans++, s[i-1]++;
11     }
12 }
13 if (s[0] == '2') ans++;    // 对于最后一位如果有进位，就还需要除以 2 一次
```