报数

直接暴力枚举显然会超时，启发我们用数学方法去做。

解法一：

正难则反，不妨反过来求前 x 个数里有多少个不会报到的数字 y，这样只要找到合适的 $x = n + y$ 即可。

根据容斥原理，有 $y = x - x/3 - x/5 + x/15$，显然 y 具有单调性，我们可以二分答案求出这个合适的 x。

时间复杂度 $O(logx)$。

解法二：

多算几个数字会发现，被报到的数字 $\bmod 15$ 的余数具有循环性。因此，可以开一个数组 $a[]$ 将 $\bmod 15$ 的余数存起来作为循环节，即 $a[] = \{0, 1, 2, 4, 7, 8, 11, 13, 14\}$。

用 f_x 表示第 x 个被报到的数字，则有：$f_x = f_{x-8} + 15$，但是仍然会超时。

进一步简化，可以推出 x 前面有多少个完整的循环节 k，然后根据数学直接计算即可得出结果。

数字游戏

想要 $a_i + b_i$ 的最大值最小，一定是 a 序列的最小值与 b 的最大值匹配，a 的次小值与 b 的次大值匹配，以此类推。

即 a 从小到大排序，b 从大到小排序，然后求解 $a_i + b_i$ 的最大值。

测试点 1~5

对于每一次询问，都重新排序 a 和 b 序列，然后打擂台求解最大值。

时间复杂度为 $O(n^2 \log n)$，预计可以得到 50 分。

测试点6~10

观察到 a_i 和 b_i 的范围很小，那么我们可以采用计数排序的方式进行处理。

每次添加元素后，分别使 $cnt1[a_i] + +$ 和 $cnt2[b_i] + +$；然后枚举 $[1, 100]$ 里的每一个数，让两个数进行配对，具体方法如下：

- 将 $cnt1[], cnt2[]$ 复制一份到 $tmp1[].tmp2[]$ 中
- 设置 i 从 A 序列的最小值 1 开始往右移动，当 $tmp1[i] \neq 0$ 时退出循环
- 设置 j 从 B 序列的最大值 100 开始往左移动，当 $tmp2[j] \neq 0$ 时退出循环
- 如果 i 和 j 没有越界，则取 $ans = \max(ans, i + j)$

- 接下来分类讨论：如果 $tmp1[i] == tmp2[j]$，则数字 i 和数字 j 的数量相同，直接抵消，$i++, j--$
- 如果 $tmp1[i] > tmp2[j]$，则 i 的数量更多，可以吃下多余的 j，$tmp1[i]- = tmp2[j], j--$
- 如果 $tmp1[i] < tmp2[j]$，则 j 的数量更多，$tmp2[j]- = tmp1[i], i++$
- 不断执行上述过程，直到 $i > 100$ 或 $j < 1$ 为止；最后输出 ans

总时间复杂度为 $O(100 * n)$。

逃生游戏

题目本质就是给出 n 个数字，每次可以删除三个数字 $x-1$、x、$x+1$，可以获得 $cnt[x] * x$ 的得分，问最大得分。

首先很容易会想到删除左右不存在的 x，或者左右只存在一个的，但是这样的思路并没有意义——因为最后还是面临如果左右数字存在如何删除的问题。

可以发现其实删除数字的顺序是无关的，那我们不妨从小到大依次删除。设 $dp[i]$ 表示数字 i 已经被处理完的情况，那么转移方式无非就是考虑之前删的是 $i-1$ 还是 $i-2$。

如果删的是 $i-2$，那么 i 就是可以删掉的；如果删的是 $i-1$ ，则 i 不能被删掉。

所以很容易得到方程：$dp[i] = \max(dp[i-1], dp[i-2] + cnt[i] * i)$

旅游计划

测试点 1~3

给暴力留的分。

测试点4~8

先通过 bfs 求出每个点到另一个点的最短路径的长度 $d_{i,j}$ ，然后暴力枚举四个点 a, b, c, d ，再求 $\max(d_{a,b} + d_{b,c} + d_{c,d})$ 就可以了。

复杂度 $O(n^4)$ 。

测试点9~10

答案就是 3，因为在完全图中，任意两点的最短路都是 1 。

测试点11~15

同样求出 $d_{i,j}$ ，然后暴力枚举三个点 a, b, c ，然后求出离 c 点最远的点 f_c，但是 f_c 可能是 a 或者 b ，所以需要求出离 c 点第二远的点 g_c，第三远的点 h_c ，这样就一定可以找出那个最远的点。

复杂度是 $O(n^3)$ 。

当然也有许多其他做法。

测试点16~20

优化上一个算法。

同样求出 $d_{i,j}$ 和 f_i, g_i, h_i 。

我们枚举两个点 a, b ，然后经过 $a \to b$ 的最远路径一定是 $f_a/g_a/h_a \to a \to b \to f_b/g_b/h_b$ 其中之一。

依次考虑这几种情况，排除掉非法的情况（经过相同的点），直接枚举就可以了。

复杂度是 $O(n^2)$ 。