1 相遇

测试点 1。和样例一样。

测试点 2。答案是 0。

测试点 3。终点在起点的左侧，只能倒着走，答案是 N。

测试点 4。向右走一格，答案是 1。

测试点 5。每次移动有 3 种方式，由于答案不超过 8，爆搜即可，搜索次数不超过 3^8。

测试点 1 ~ 10。将爆搜改为 BFS 即可，注意对状态的处理，由于限制了 N 和 K 都在 0 到 10^5 之间，因此在 BFS 时候可以只保留一个区间$[L, R]$内的状态，比如 $L = -10^6$，$R = 10^6$。当然，存在更紧的上下界 L、R，取$\pm 10^6$ 已经足够了。

2 求和

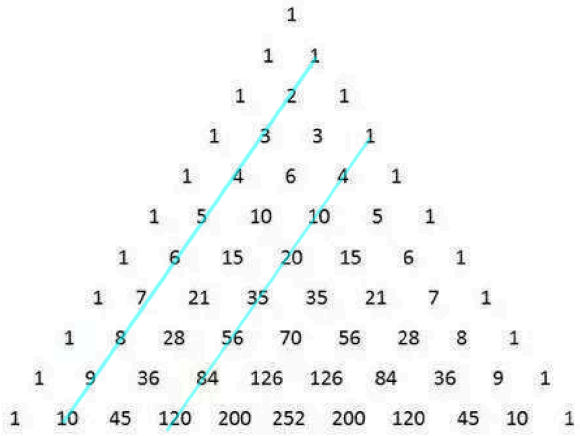
测试点 1～2。考察是否读懂题目，只需要模拟一次操作，期望得分 8 分。

测试点 1～8。考察是否读懂题目，正确处理取模操作，期望得分 32 分。

测试点 1～16。实际上，每次操作可以等价于一次矩阵乘法操作，转移矩阵是一个左下角全为 1，对角线全为 1，右上角全为 0 的矩阵。使用矩阵快速幂计算，时间复杂度 $O(n^3 \log k)$，期望得分 56～64 分。在计算矩阵乘法 $C=AB$ 的时候，朴素的计算方法为 $c[i][j] \mathrel{+}= a[i][k] * b[k][j]$。这里我们将矩阵 B 进行转置，计算公式变为 $c[i][j] \mathrel{+}= a[i][k] * b[j][k]$。按照 i，j，k 顺序循环，可以使得程序的“访存局部性”变好（涉及程序的底层优化机制），有一定可能多通过一些测试点。

测试点 1～20。利用这个转移矩阵的对称性，可以将矩阵乘法时间复杂度降低到 $O(n^2)$，期望得分 80 分。

测试点 19～22。这部分测试点用来提示选手想出满分算法。当 $a_i=1$ 时，初始序列为(1, 1, 1, 1, 1, 1, …)，第 1 次操作变为(1, 2, 3, 4, 5, 6, …)，第 2 次操作变为(1, 3, 6, 10, 15, 21, …)，第 3 次操作变为(1, 4, 10, 20, 35, 56, …)，第 5 次操作变为(1, 5, 15, 35, 70, 126, …)。



可以看到，每次操作相当于沿着杨辉三角向右下方移动一格。因此最终结果可以直接用组合数表示出来，这个组合数分子和分母的元素个数都是 $O(n)$个，计算分子每个元素的乘积 x，再计算分母每个元素的乘积 y，最后用 x 乘 y 的逆元便可以求出这个组合数。

测试点 1 ~ 25。考虑 $n = 4$ 的情况，进行 1 次操作后序列变为：

a_1

$a_1 + a_2$

$a_1 + a_2 + a_3$

$a_1 + a_2 + a_3 + a_4$

进行 2 次操作后序列变为：

a_1

$2a_1 + a_2$

$3a_1 + 2a_2 + a_3$

$4a_1 + 3a_2 + 2a_3 + a_4$

进行 3 次操作后的序列变为：

a_1

$3a_1 + a_2$

$6a_1 + 3a_2 + a_3$

$10a_1 + 6a_2 + 3a_3 + a_4$

可以观察到每个元素都是由最开始的 n 个元素加权得到的，当操作 k 次时，加权系数恰好是 $a_i = 1$ 时操作 $k - 1$ 次所得结果的序列反过来。比如上面的例子，共操作 3 次。对(1, 1, 1, 1)操作两次可以得到(1, 3, 6, 10)，反过来即可得到(10, 6, 3, 1)。在测试点 19 ~ 22 的基础上加上加权这一步即可。时间复杂度 $O(n^2 + n\log P)$，其中 P 为模数，每次计算逆元的时间复杂度为 $O(\log P)$。

3 小乔

测试点 1～2。简单计算就可以知道答案，期望得分 10 分。

测试点 1～6。我们可以把坐标系划分成 $m \times r$ 块，对于一个扇形，将它覆盖到的每一块的覆盖次数加 1，最后统计有哪些块的覆盖次数不小于 k，计算面积即可。复杂度 $O(nmr)$，期望得分 30 分。

测试点 7～12。这部分测试点所有扇形的半径都相同，这样我们可以把扇形“拉直”成线段，覆盖的区域就成了数轴上的区间。那么问题就转化成了经典的区间覆盖问题。我们把每个区间拆成左端点和右端点两个位置，然后把所有位置按照坐标排序。从左到右扫，遇到左端点则将覆盖次数加 1，遇到右端点则减 1。当覆盖次数不小于 k 时，把这个区间的面积加入答案。复杂度 $O(n\log n)$，结合前面的算法可以得到 60 分。

测试点 1～20。我们把每个扇形的左右段对应的角度都提出来排序，将圆周划分成 $O(n)$个部分。我们考虑一个区间，取出所有覆盖了这个区间的扇形。在这个区间内，半径较大的扇形一定完全覆盖了半径较小的扇形。那么这里被覆盖了至少 k 次的面积就是半径第 k 大的扇形在这个区间内的面积。

如果我们按照角度顺序依次处理每个区间，那么我们就需要一种数据结构来帮助我们快速查找第 k 大的元素，还需要支持插入和删除一个数。一个可行的，也是标程采用的数据结构是平衡树。当然这不是唯一的做法。另外，还有 20 分的数据 $k = 1$，可以直接用堆维护。使用平衡树的算法复杂度 $O(n\log n)$，期望得分 100 分。