练习地址:

https://www.nowcoder.com/acm/contest/206#description

birthday:

思路：考虑费用流时把每个 part 拆成 n 个点，选择第 i 个点的代表为放置 i 块蛋糕和(i - 1)块蛋糕的时间差，这个时间差是递增的，因此在费用流的过程中必定会从小到大选择

具体建图：左边 n 个点代表 n 个蛋糕，右边 m * n 个点代表 m 个 part,每个 part 拆成 n 个点。源点向每个左边的点连一条流量 1 费用 0 的边，每个右边的点向汇点连一条流量 1 费用 0 的编。每个蛋糕向可以放的两个 part 的所有点连边，连向第 i 个点的费用为 i^2 - (i - 1)^2，流量为 1。这样求最小费用流既为答案。

board:

把格子 N 染色，第 i 行第 j 列格子的颜色为(i + j) % N。那么每次操作时，必定是 N 种不同的颜色都有一格被操作到，因此最后任何颜色格子的和必定是相等的。因此只需要记录每种颜色格子的和，并算出缺失格子的颜色 C，用其余颜色的和减去颜色 C 的和即可

Circle：

因为(i,i+1)=1 且(1,n)=1，所以把 1...n 依次放进一个环，就可以啦。答案为 n。

Growth：

把奖励的 x 拿出来从小到大排序，得到 x1,x2,...,xn。

把奖励的 y 拿出来从小到大排序，得到 y1,y2,...,yn。

用 $v[i][j]$表示 a 值到达 x_i，b 值达到 y_i 时接下来每天可以得到的奖励。

$v[i][j] = v[i-1][j] + v[i][j-1] - v[i-1][j-1] + t[i][j]$

其中 $t[i][j]$为满足 $x=i$，$y=j$ 的奖励的总和。

用 $f[i][j]$表示 a 值达到 x_i，b 值达到 y_j 时已经拿到的奖励的最大值。

$f[i][j] + (x[i+1] - x[i] - 1) * t[i][j] + t[i+1][j] \to f[i+1][j]$
$f[i][j] + (y[j+1] - y[j] - 1) * t[i][j] + t[i][j+1] \to f[i][j+1]$

最后统计一下答案就可以了。

kingdom:

$f[i]$代表 i 个点时的答案，$g[i][j]$代表若干颗树加起来，size 和为 i，每棵树

size<=j 时，这些树的代价和最大是多少

从 1 到 n 枚举 i，在 i 固定时枚举心腹的影响力大小更新 $f[i]$，然后用类似背包

的思路更新 $g[i][1]$~$g[i][i]$

复杂度 $O(N^2)$

Matrix：

w 个格子的重心的坐标为（$\sum x_i*w_i / \sum w_i, \sum y_i*w_i / \sum w_i$)。

那么其实我们只要维护$\sum x_i*w_i$，$\sum y_i*w_i$，$\sum w_i$ 就可以了。

假设我们现在有一个顶点为(x, y)的三角形，我们想要推到顶点为(x, y+1)的三

角形，观察两者之间的差异，会发现在推过去的过程中，其实就是删去了一个

斜条，又加入了一个斜条。

同理，从(x, y)到(x+1, y)其实只是删去了两个斜条，加上了底上的横条，而这

些关键的值都是可以通过前缀和的方法维护。

Mountain：

考虑山中最高的一座，最优操作一定是从第一座山的左下角开始不停地往上爬，然后从最高的山不停地往下爬爬到最后一座山的右下角。

所以答案为最高山的高度*2。

清明梦:

首先每条路径从 LCA 处分开可以拆成两条链

假设链 A->B 执行了第 i 次染色操作，假设 A 是 B 的祖先，那么我们在 B 点加入一个"插入 i"的事件，在 A 的父亲点加入一个"删除 i"的事件

然后 dfs 整颗树求解，每个点维护一个线段树。处理一个点时先合并所有儿子的线段树，然后再处理这个点上的事件，得到线段树之后询问第 K 大值既可得到答案。

复杂度分析：

```
Node* merge(Node* a, Node* b) {
    if (a == NULL) return b;
    if (b == NULL) return a;
    a->sum += b->sum;
    a->child[0] = merge(a->child[0], b->child[0]);
    a->child[1] = merge(a->child[1], b->child[1]);
    return a;
}
```

考虑以上的线段树合并，每次合并会减少一个区间。而在事件点插入、删除的时候会产生至多 log 个区间，因此复杂度为 O(NLogN)

最短路：本题十分直接。我们不断地把度数为 1 的点删掉，把度数为 2 的点收缩，最后会得到一个图，和原图的点数与边数之差相同，且新图中每个点的度数都至少是 3。这就是说我们会得到一个 200 个点 300 条边以内的图。新图可

以用 Floyd 算法预处理所有点对之间最短路。询问时，将询问转化到新图上即可。转化时需要注意细节。

排序：设 m 是 a 和 b 的差的 lowbit。我们先假设 m 是 1，即 a 和 b 的奇偶性不同。这时通过适当的构造，我们可以用常数步交换任意两个奇偶互异的数字的位置。交换奇偶相同的数只要借一个和它们奇偶不同的数即可。如此我们便可交换任意两个数，即此时没有无解。下面考虑 m 大于 1。我们称一个数字 x 的低位为(x&(m-1))，高位为 x 减去它的低位。我们发现数组的低位是无法利用交换魔法的，只能用到加法和异或。也就是说，我们必须用加法和异或排好数组的低位。排好低位后，我们按照低位将所有数字分为若干组，每组内（和之前 m 为 1 的情况类似）是没有无解的。现在问题只剩如何用加法和异或排好低位。可以发现，a[0],…,a[m-1]的低位和 a[m],…,a[2m-1]的低位必须完全一致且均为 0 到 m-1 的一个排列，否则它们无法同时通过加法和异或排好。同理 a[2m],…,a[3m-1]的低位也必须一致。所以，我们只要用加法和异或排好 a[0],…,a[m-1]即可。这其实是本题的 n=m 且没有交换魔法的版本（因为现在超过 m-1 的加法和异或是没用的）。在这个版本下，从升序排列只能生成 (m/2)*(2**(m/2))种不同的排列。这些排列可以用递归法构造：m=2 时用一次加法即可（后面会解释为什么用加法不用异或）。m 更大时，我们发现连续使用加 1 和异或 1 可以达到奇数都加 2，偶数都不变的效果。这个操作实际相当于只考虑奇数且不考虑个位情况下的加 1 操作。这就可以提取所有奇数做递归，将所有奇数排列成任意的 m/2 时的可能排列。同理，先异或 1 再加 1 起到的是偶数加 2 奇数不变的效果。我们相似地递归偶数部分。注意到，递归一侧时，

异或操作是会影响另一侧的。奇数和偶数两侧的异或操作的异或和必须相同（因为整体考虑，异或操作其实是同时作用在奇数和偶数上的！），除了个位和最高位。（除了个位是因为我们根本不考虑个位，个位有值的异或操作只用在异或 1 的时候了。除了最高位是因为最高位异或可以用奇数+2 操作模拟出来。）这样可以得到的排列数最多为(m/2)*(2**(m/2))，同时满足条件的排列都可以用递归法构造出来。

土龙弟弟：下面会有一些定义。出此题是期望比赛的时候大家凭感觉得到结论，不证明。

首先题目问的是在 torus 上扣若干个洞，所得曲面上的闭环的同伦等价类。（定义：torus：就是我们定义的地图，形似甜甜圈。闭环：就是说土龙弟弟一天走的路径。因为会回来所以叫闭环。同伦等价类：就是一个土龙弟弟的路线可能会变换。凡是可能属于同一个土龙弟弟的两条路线就是相互同伦的。题目问的就是最少能分成多少组使得组内相互同伦。）我们把扣掉的洞用线连起来，使得线之间不相交且按照线将曲面分割成若干简单的部分。（简单的部分：就是这一部分是你在纸上随便画个不自交的闭环得到的东西。简单是因为这一个区域中所有的闭环都同伦，因为可以缩小到很小再移动到一起。）分割区域这一步需要小构造一下。大体思路就是每一行都画一条横的线，这线实际会被洞分割成多条线。然后有些区域是个环（即从左边走到最右边再向右回到最左边），即并非简单区域。这些区域再画一条竖线。

这样分割后，一个闭环就可以用依次穿过的线的编号组成的字符串来表示（除了记录哪条线，方向也有用。并且由于是环，字符串也是循环的。）然后消去

相邻相反括号（即连续的正穿和反穿同一条线的部分。因为每个区域都是简单的，所以连续正穿和反穿中间的部分可以不断收缩直到不穿过这条线，所以消去后所得的曲线和之前是同伦的。）最后为了判断循环串相等要用最小表示法。证明：我们要证的是同伦当且仅当消去相邻相反括号后的序列循环相等。假设 a 经过消括号得到（不能再消的）b，显然 a 和 b 是同伦的，因为每一步都同伦。假设 a'消括号得到 b'。如果 b 和 b'循环相等,a 和 b 同伦。这就证明了一半。反过来，对一个曲线做同伦变换时，只能产生或消除相邻的相反括号。所以如果 b 和 b'不等，他们又都没有相邻相反括号可以消去，它们就不可能通过产生和消除相邻相反括号相互转换，即 b 和 b'不同伦，即 a 和 a'不可能同伦。

最后注意没有洞是特殊情况。题面里除去了这种情况。其实 torus 上扣若干洞后的基本群总是 free group，但是没扣洞时的基本群是 Z^2（平面整数格点加法群）。