# A. 按位或

## 子任务 1

$f(i,j)$ 表示前 $i$ 个数字或起来为 $j$ 的方案数，枚举第 $i$ 个数字是多少即可转移。

时间复杂度 $\mathcal{O}(nt^2)$。

## 子任务 3

定义 $x$ 的子集为所有与 $x$ 或起来为 $x$ 的数。

发现我们可以容易计算 $n$ 个数或起来为某个数的子集的方案数，所以可以容斥 $t$ 计算。

时间复杂度 $\mathcal{O}(t^2 \log n)$。

## 正解

现在我们要快速计算 $x$ 的子集中为 $3$ 的倍数的数的数量。

发现 $2^i \bmod 3$ 的结果只有 $1$ 和 $2$ 两种，我们可以把二进制下的每一位分到其中一类中。

因此枚举容斥中 $t$ 的位中 $\bmod 3$ 为 $1$ 和 $2$ 的位各有多少。

$f(i,j)$ 表示做到第 $i$ 位，此时 $\bmod 3$ 为 $j$ 的方案数。

时间复杂度 $\mathcal{O}(\log^2 t \times (\log t + \log n))$。

# B. 最短路径

## 正解

假设回到起点，那么路径长度就是这 $k$ 个点的虚树的边长度之和乘 $2$，那么减去虚树的最长链就是最短路径长度了。

所求的期望就可以转化为求**虚树边长之和的期望**和**虚树直径的期望**。

求边长之和的期望只要枚举每条边，它出现在虚树的概率就是这条边两边都有小饼干的概率。

求虚树直径的期望可以暴力一点。钦点一棵树中的某个点对 $(u,v)$ 为直径，当且仅当 $\text{dis}(u,v)$ 最长且字典序最小，并且钦点 $u<v$，那么直径是唯一的。枚举所有点对 $(u,v)(u<v)$，考虑这条路径是直径的概率。

首先，考虑点对 $(u,w)(w\neq v)$ 和 $(v,w)(w\neq u)$ 的影响，即先保证不会出现其他以 $u$ 或 $v$ 的端点的路径为直径。那么我们发现，对于一个点 $w$，它不能出现当且仅当下面**几个条件之一被满足**：

- $\text{dis}(u,w) > \text{dis}(u,v)$
- $\text{dis}(u,w) = \text{dis}(u,v) \land w < v$
- $\text{dis}(v,w) > \text{dis}(u,v)$
- $\text{dis}(v,w) = \text{dis}(u,v) \land w < u$

不难发现，假设存在 $(x,y)$ 比 $(u,v)$ 更优，并且这两条路径端点不重合，那么 $x$ 或 $y$ 一定会满足上面几个条件之一。所以这样我们也考虑到了其他点对的情况。因此上面的条件是**充分且必要**的。

我们数出能出现的点数 $\text{cnt}$，算概率就是算其他 $k-2$ 个小饼干都落在这 $\text{cnt}$ 个里面的概率。

# C. 仙人掌

注意模数是 $993244853$。

## 子任务 1

枚举排列算行列式。

## 子任务 2

高斯消元算行列式。

## 子任务 4

考虑行列式的定义式中枚举的排列，将排列分解成不相交的轮换。如果排列对应的贡献不为 $0$，那么对应到图上就是若干个不相交的环（或者匹配边），但是环的并集要是所有点。

对于一棵树，不需要考虑环的情况，可以树形 DP 处理。

## 正解

那么在仙人掌上就是对应着，可以选整个环，也可以选边，要把所有点集都恰好覆盖一次。选环的贡献是 $2\times(-1)^{\text{环长}-1}$，选边的贡献是 $-1$。

在仙人掌上 DP 即可，时间复杂度 $\mathcal{O}(n)$。

# D. 对弈

## 子任务 1

$n\le 5$ 直接枚举+分类讨论即可。

期望得分 $10$ 分。

## 子任务 3

$k=2$ 的部分，我们不难发现，如果两个棋子开始是紧靠的，那么先手只能往左移，后手总能移动同样的距离使得仍然是紧靠的，这样后手就必胜。否则先手可以移到紧靠的位置，先手必胜。

## 子任务 4/5

可以得到一个性质：在最优策略下，如果可以，$\text{Alice}$ 只会向右移动，$\text{Bob}$ 只会向左移动。

> **证明：** 为了证明这个性质，记 $a_i$ 表示第 $i$ 个红棋的位置，$b_i$ 表示第 $i$ 个蓝棋的位置，我们将一个状态表示为一个 $\frac{k}{2}$ 元组 $(b_1-a_1-1,b_2-a_2-1,\cdots,b_{\frac{k}{2}}-a_{\frac{k}{2}}-1)$。记 $d_i=b_i-a_i-1$，如果 $\text{Alice}$ 将 $a_i$ 向左移动，那么在它右边的那个蓝棋 $b_i$ 一定可以移动相同距离，使得对应的 $d_i$ 不变，但是显然 $a_i$ 的移动范围变小了，不这么做显然不会更劣。

所以 $\text{Alice}$ 只会向右移动，$\text{Bob}$ 只会向左移动。这相当于有 $\frac{k}{2}$ 堆石子，第 $i$ 堆的石子个数为 $d_i$，每次可以选 $1\sim m$ 堆，从这几堆中各取走若干石子，每堆至少取一个。

不难发现 $(0,0,\cdots,0)$ 是必败态，因为此时轮到的人无论怎么走，而另一个人总能使其回到这个 $(0,0,\cdots,0)$ 的状态。

因此相当于轮到某人时，所有石子都已经被取完了，这个人就输了。

$m=1$ 的时候就是经典的 NIM 游戏，当前状态必败，当且仅当所有石子个数的异或和为 $0$。

根据 DP 的优劣，可以通过子任务 4 或子任务 5。

## 正解

实际上 $m \geq 1$ 的问题是经典的 NIM-K 问题。

考虑简化后的问题，有 $n$ 堆石子，每堆有 $a_i$ 个，每次可以选 $1 \sim m$ 堆，从这几堆中各取走若干石子，每堆至少取一个。若轮到某人时，所有石子都已经被取完了，这个人就输了。

结论是，当前状态必败，当且仅当将所有 $a_i$ 转成二进制后，每一位 $1$ 的个数 $\bmod (m+1) = 0$。

> **证明：**
>
> 1. 全为 $0$ 的局面一定是必败态。
> 2. 任何一个 $\mathrm{N}$ 状态（必败状态），经过一次操作以后必然会到达 $\mathrm{P}$ 状态（必胜状态）。
>    在某一次移动中，至少有一堆被改变，也就是说至少有一个二进制位被改变。由于最多只能改变 $m$ 堆石子，所以对于任何一个二进制位，$1$ 的个数至多改变 $m$。而由于原先的总数为 $m+1$ 的整数倍，所以改变之后必然不可能是 $m+1$ 的整数倍。故在 $\mathrm{N}$ 状态下一次操作的结果必然是 $\mathrm{P}$ 状态。
> 3. 任何 $\mathrm{P}$ 状态，总有一种操作使其变化成 $\mathrm{N}$ 状态。**从高位到低位**考虑所有的二进制位。
>    假设用了某种方法，改变了 $k$ 堆，使第 $i$ 位之前的所有位的 $1$ 的个数都变成 $m+1$ 的整数倍。现在要证明总有一种方法让第 $i$ 位也变成 $m+1$ 的整数倍。显然，对于已经改变的那 $k$ 堆，当前位可以自由选择 $1$ 或 $0$。
>    设除去已经更改的 $k$ 堆，剩下的堆第 $i$ 位上 $1$ 的总和 $\bmod (m+1) = sum$，分类讨论：
>    (1) $sum \leq m-k$，此时可以将这些堆上的 $1$ 全部拿掉，然后让那 $k$ 堆的第 $i$ 位全部置成 $0$。
>    (2) $sum > m-k$，此时我们在之前改变的 $k$ 堆中选择 $m+1-sum$ 堆，将他们的第 $i$ 位置成 $1$，剩下的置成 $0$。由于 $m+1-sum \leq k$，故这是可以达到的。

那么就可以 DP 了，设 $f(i,j)$ 表示考虑了二进制的前 $i$ 位，并且当前总石子数量为 $j$ 的方案数。转移的时候只要枚举这一位 $1$ 的个数，然后乘上一个组合数系数即可。

时间复杂度 $T(n) = \sum_{i=0}^{\lfloor \log_2 n \rfloor} \frac{nk}{2^i} = \mathcal{O}(nk)$。