THU DDF Contest Hints

A. Random Numbers

这个题的直观想法就是我们可以先估计出一个 m_0 来，然后在这个值的上下枚举 m。现在不妨假设已经知道了 m，那么根据 $\sum_{i=1}^{n} a_i + n \cdot k \equiv \sum_{i=1}^{n} b_i \pmod{m}$，可以解出 $\gcd(n, m)$ 个可行的 k。现在知道了 m 和 k，很多人可能做过最近某场的某个 cf 题，类似那个直接暴力 check 若干项就好了。可以证明这个上下枚举次数是 $O(\frac{m}{n})$ 的。然后暴力 check，每次 check 不对的概率是 $\frac{n}{m}$ 的，于是这样暴力期望上能在 $O(1)$check。

B. Defense Tower

每个控制自己的防御塔一定控制的是一个连通快，对每个连通快用一个双向链表依照 dfs 序维护相邻的割边。

每次增加后，暴力地从周围的连通块“掠夺”节点。具体实现时，通过 $O(1)$ 的 Level Ancestor 得到新的割边，那么“掠夺”来的连通块的割边一定是旧割边和新割边之间的边。因为题目的要求，每次“掠夺”时，被“掠夺”点的 eff 值都会增大，所以一定会向周围继续“掠夺”，所以可以暴力地抠出这个连通块的割边。如果知道一个连通快的所有割边，那么用总数减掉每个割边外的连通快大小，就是当前连通块的数量，因此计算答案。这里微妙的地方是 eff 值相等的情况，可以考虑在每条边中间加一个虚拟点，复杂度分析一样奏效。

因为同一时刻只有 $O(\sqrt{n})$ 个 $\geq \sqrt{n}$ 的块，而一个 $\leq \sqrt{n}$ 的块最多被掠夺 $O(\sqrt{n})$ 次，所以总复杂度是 $O(n^{1.5})$.

C. Eulerian Orientation

首先，图的欧拉子图的数量是 $2^{|E|-|V|+c}$，其中 c 是连通块的数量。

设边集是 E，选取的边集是 $F \subseteq E$, 那么 $\mathbb{E}[|F|] = \mathbb{E}[\sum_{e\in E}\sum_{f\in E}[e \in F][f \in F]] = \sum_{e\in E}\sum_{f\in E}\Pr[(e \in F) \wedge (f \in F)]$.

首先 e, f 都不是割边，如果 $e = f$，那么包含 e 的欧拉子图的数量是 $2^{|E|-1-V+c}$. 如果 $e \neq f$，而且 $\{e, f\}$ 不是割集，那么包含 e, f 的欧拉子图的数量是 $2^{|E|-2-V+c}$。如果 $\{e, f\}$ 是割集，那么数量是 $2^{|E|-1+V+c}$.

所以，只需要统计 $\{e, f\}$ 是割集的数量。一个方便的方法是随机化，给每条非树边一个 $[0, 2^{64})$ 的随机权值，树边的权值设为覆盖它的非树边的权值的异或。那么 $\{e, f\}$ 是割集当且仅当 e, f 的权值相等。

D. Tube Master II

逐行逐列确定每个格子的边，通过一些分类讨论，就知道只要确定了每个点左、上，和右上的边的状况，那么就可以唯一确定这个点剩下的（右、下）两条边的状况。

E. Palindrome

做这个题首先需要知道三个结论（可能只知道第一个也够了）：

1. 一个回文串的所有回文后缀的长度可以由 $O(\log n)$ 个等差数列表示。具体是这样的：假设 $l_0, l_1, l_2, ..., l_m$ 是回文串 s 的 m 个回文后缀且 $0 = l_0 < l_1 < l_2 < \cdots < l_m$。令 $d_i = l_i - l_{i-1}$，如果 $d_i = d_{i-1}$ 那么就把 l_i 和 l_{i-1} 归在同一个等差数列里面。可以证明这样只会产生 $O(\log n)$ 个等差数列，具体证明参考 Mikhail Rubinchik 关于EERTREE的论文。
2. 令 $l_1, l_2, ..., l_m$ 是 1 中的某个等差数列，令公差为 d，那么一定有 $s(l_1, d) = s(l_2, d) = \ldots s(l_{m-1}, d) \neq s(l_m, d)$，其中 $s(x, d)$ 表示 s 从 x 位置开始往后取 d 个字符得到的子串。证明很容易，略去。
3. 给出两个字符串 s 和 t，已知 $s(0, d) = s(d, d) = s((k-1)d, d) \neq s(kd, d)$，那么 t 和 $s(0,), s(d,), s(2d,), \ldots, s(kd,)$ 的最长公共前缀是一定是先不变，然后变化一下，之后以公差为 d 递减，其中 $s(x,)$ 是以 x 开始的后缀。证明应该也很容易，略去。

回到原问题，如果会做两个区间，那么多个区间也是类似的，不妨只考虑两个区间。如果只有两个区间，答案的贡献来自于 2 部分：

1. 两个区间本身内部的回文串数目，这个是经典问题，离线 + 线段树就可以搞定，具体做法参考这里。
2. 左端点在第一个区间，右端点在第二个区间的回文串，其实根据回文中心的位置还可以细分为 2 部分。要计算这部分的贡献，需要解决下面的问题：一直两个串 s 和 t 的回文前缀和回文后缀，如何求出 st 的所有回文后缀。

可以发现，这些回文后缀来源于两个地方：

1. 回文中心在 t 中：一部分是 t 本来的回文后缀，另一部分是 t 的某些回文前缀扩展得来。
2. 回文中心在 s 中：这部分是 s 的某些回文后缀扩展而来。

先考虑第一种，令 $t = uv$，u 是一个回文串，那么只要 s 的某个后缀是 v^R（v 的翻转），那么就获得了一个回文后缀。根据最上面的三个结论，我们可以把 t 的一坨回文前缀一起处理。每坨回文前缀都是结论 1 中的一个等差数列，然后判断某个回文前缀能否扩展其实就是求 s^R 和 $t(x,), t(x+d, d), \ldots, t(x+(k-1),), t(x+kd,)$ 求最长公共前缀。根据结论 2 和 3，我们就可以在 $O(1)$ 时间内知道哪些回文前缀是可以扩展的（显然是一些连续的回文前缀）。

第二种也是类似的，不过是变成了 t 和 $s^R(x,), s^R(x+d,), s^R(x+2d,), \ldots, s^R(x+(k-1)d,), s^R(x+kd,)$ 求最长公共前缀而已。

事实上，在求得最长公共前缀的时候，我们已经可以计算新增的回文串的个数了，直接利用结论 3 就好了。于是我们已经会合并两个区间，以及计算两个区间的答案了，多个区间也是一样的过程。

还剩下一个问题，获取原始区间的所有回文前缀和回文后缀。这个可以直接套用 EERTREE，或者用上述过程一个一个字符往后加，同时维护出当前的回文后缀。

中间细节可能比较多，在纸上好好画画应该就能搞定。复杂度 $O((n + \sum k_i) \log n)$。

F. Median on Binary Tree

对于固定的 a，枚举 x，考虑答案是否能 $\geq x$，如果把 $\geq x$ 的当做 $+1$，$< x$ 的当做 -1，那么等价于问有没有子树的和 $\geq a$。

可以设 $f(i, \{0,1\})$ 表示点 i 所在的点选或不选的最大子树。如果我们从大到小枚举 x，那么点的权值会从 -1 变到 $+1$，因为树高是 $O(\log n)$ 的，可以每次暴力更新祖先的 f 值，当最大的 f 时第一次变成 a 时，所对应的 x 就是答案。

G. Card Shuffle

对于 $(1, 2, \dots, k, x)$ 的前缀，在 2^k 步后 x 会往后走 $k - x$ 步，其余不变。那儿等价于前缀不变，后缀减去 k 后递归做。

所以只要考虑 1 到第一位所需要的步数，假设 1 在 t 位置需要的步数是 $f(t)$，并假设 1 前面没有 0. 那么只有 $2, 3, \dots, (t-1)$ 共 $(t-2)$ 个数 $< t$。所以至少有 2 个数 $\geq t$，也就是说在前 $(t-2)$ 个位置有 $\geq t$ 的数，而且它到第一位之后会让 1 往前一位，所以 $f(t) \leq f(t-1) + f(t-2)$.

H. Independent Events

考虑泰勒展开 $\ln(1-x) = -\sum_{i=1}^{\infty} \frac{x^i}{i}$，那么就可以用线段树维护，展开 20 项差不多就够了。

I. Territory Game

首先，如果 Alice 和 Bob 在 k 不内不可达，那么显然答案是 $k \bmod 2$。接下来按照 k 的奇偶性分类讨论。

考虑 k 是偶数的情况，那么答案只可能是 0 或者 -1。

- 如果 A 和 B 之间有奇数条边，如果两个人对着走互怼的话，那么肯定两个人各占的点是一样的。那么对于 Alice 来说，只愿意这么走，因为她不能吃掉 Bob 一个点。
- 如果 A 和 B 之间有偶数条边，如果互怼的话，Bob 会比 Alice 多一个点，于是 Alice 的策略就是逃跑，也就是找到一条长度为 $\frac{k}{2}$ 的简单路径，不经过 Bob 的势力范围（Bob 在 $\frac{k}{2}$ 步内能够到达的所有点）。

如果 k 是奇数的话，如果 Alice 和 Bob 互怼，Alice 总是可以多吃掉一个点。那么 Bob 的策略和上面的类似。如果他们之间有偶数条边，Bob 总是可以做到只被多吃一个点，直接互怼就好了。如果他们之间有奇数条边，为了防止被多吃一个点，那么 Bob 的策略就是逃跑。

剩下来就还剩下一个问题，如何判断能够逃跑：给出两个点 A 和一条边 e，求出从 A 出发不能经过 e 的最长简单路。

可以观察到最长路径一定有一个直径端点，于是只要 dp 出，删掉每条边之后分成的两个子树的直径就可以了、