T1 天才俱乐部

- 10pts

容易发现 $k > \max\limits_{i=1}^{n} a_i$ 时取模没有意义，所以枚举 $k \in [1, \max\limits_{i=1}^{n} a_i]$ 然后暴力 $O(n)$ check。

总时间复杂度为 $O(tn\,a_i)$。

- 100pts

考虑模的定义，有 $x \bmod y = x - \lfloor \frac{x}{y} \rfloor y$。

所以 $s = \sum\limits_{i=1}^{n} (a_i \bmod k) = \sum\limits_{i=1}^{n} (a_i - \lfloor \frac{a_i}{k} \rfloor k) = (\sum\limits_{i=1}^{n} a_i) - k \sum\limits_{i=1}^{n} \lfloor \frac{a_i}{k} \rfloor$，即 $k \sum\limits_{i=1}^{n} \lfloor \frac{a_i}{k} \rfloor = (\sum\limits_{i=1}^{n} a_i) - s$，也就是 $k \mid (\sum\limits_{i=1}^{n} a_i) - s$。

直接枚举 $(\sum\limits_{i=1}^{n} a_i) - s$ 的所有因数，再暴力 $O(n)$ check。

总时间复杂度为 $O(tn\sqrt{\sum\limits_{i=1}^{n} a_i})$。

T2 实战教学

- 10pts

暴力枚举配对方案即可。

总时间复杂度为 $O((2n)!)$。

- 100pts

看到最小化最大值首先想到二分答案，考虑如何check。

假设当前要check的答案为 x。按 a_i 从大到小考虑，与它配对的必然有 $b_j \le x - a_i$。容易发现，a_i 越大，限制就越紧，即符合限制的元素越少。我们按 a_i 从大到小考虑，限制是越来越宽松的，符合 a_i 限制元素的集合必然是符合 a_{i+1} 限制的集合的子集，所以我们在当前集合中选择时不需要考虑对后续集合的影响。还是因为 a_i 越大限制就越紧，所以我们如果能提前选掉 a_i 大的元素，后面选择的时候就更可能有解。具体来说就是我们在符合当前限制的所有元素中选择 a_j 最大的元素，可以用一个set维护。

总时间复杂度为 $O(n \log V \log n)$。

T3 穿越银匙之门

- 10pts

直接暴力枚举每次的操作。

可能需要稍微注意一下实现的常数。

总时间复杂度为 $O(tn(2n)^n)$。

- 40pts

发现无根树不好办，要是存在一个根就好了。

假设是有根树，我们可以得出一些限制。每个点被操作当且仅当它在 A 和 B 中的父亲不一样；如果一个点不被操作，但是父亲被操作了显然无解；一个点操作一定比其父亲更早。用拓扑排序随便搞一下就好了。

考虑如何把无根树变成有根的。发现根的本质是没有被操作，所以我们枚举一个点，钦定其不被操作作为根即可。

总时间复杂度为 $O(tn^2)$。

- 100pts

但是还有可能所有点都被操作过（即答案为 n），因为一个点不能被操作两次，所以我们枚举第一次操作，然后把被第一次操作的点作为根即可。

总时间复杂度为 $O(tn^3)$。

T4 绳网委托

- 10pts

发现操作的左右端点在连续段的中间显然不优，每段可以被视作一个整体，然后暴力枚举操作后的序列即可。

考虑得到一个操作后的序列最小操作次数，从左往右扫，如果不对就直接把对的翻转过来。

总时间复杂度为 $O(n!n^2)$。

- 30pts

下文LIS都指最长不降子序列。

若 $a_i = 0$ 则赋权为1，若 $a_i = 1$ 则赋权为-1。则一个01序列的LIS就是 $\text{cnt}_1 + s_x$，其中 cnt_1 是整个序列中 $a_i = 1$ 的个数，s 是权值的前缀和数组，x 是LIS的分界点（LIS是000……0111……1的形式，分界点即最后一个0的位置）。

所以我们需要找操作后序列的一个权值和最大的前缀。发现操作 k 次后的前缀，来自于一个操作前的前缀和 k 个操作前的不交区间。因此我们要从序列中选出一个前缀和若干不交区间，使得权值和尽可能大，可以直接DP。

总时间复杂度为 $O(n^3)$。

- 100pts

考虑优化找权值和最大的不交区间的过程。

贪心地找一个权值和最大的区间，然后把其中的权值全部变为原本的相反数，这样下次再取到这里相当于扔掉了之前取的部分。贪心的过程用线段树维护。

总时间复杂度为 $O(n \log n)$。