1、**于等(slauqe)**

注意到实际上只有 4 种位置：

- $a_i = 1, b_i = 1$
- $a_i = 1, b_i = 2$
- $a_i = 2, b_i = 1$
- $a_i = 2, b_i = 2$

~~好像就能二进制优化 $n^2 \log n$ 子~~

注意到 a, b 可以交换，不妨设 $\sum a_i \geq \sum b_i$。

以下记 $S = \sum a_i$，$T = \sum b_i$，$cnt_{p\in\{1,2\},q\in\{1,2\}}$ 为有多少个位置 $a_i = p$ 且 $b_i = q$。

那么我们有 $S = cnt_{1,1} + cnt_{1,2} + 2cnt_{2,1} + 2cnt_{2,2}$，$T = cnt_{1,1} + 2cnt_{1,2} + cnt_{2,1} + 2cnt_{2,2}$

相减得到 $S - T = cnt_{2,1} - cnt_{1,1}$。

而我们要构造一个子集满足 $\sum a_i = \frac{1}{2}S$，$\sum b_i = \frac{1}{2}T$，而 $\sum a_i - b_i = \frac{1}{2}(S-T)$。

记 $cnt'_{p\in\{1,2\},q\in\{1,2\}}$ 为选中的子集中 $a_i = p, b_i = q$ 的数量。同理得到 $cnt'_{2,1} - cnt'_{1,2} = \frac{1}{2}(S-T)$。

所以我们至少要选 $\frac{1}{2}(S-T)$ 个 $a_i = 2, b_i = 1$，而这显然是能选出来的，因为 $a_i = 2, b_i = 1$ 的个数至少是 $S - T$。

选出来这 $\frac{1}{2}(S-T)$ 个 $a_i = 2, b_i = 1$ 之后，相当于我们再选一些位置，使得选中位置 a 的和等于 b 的和等于某个值。而我们选一个 $a_i = 2, b_i = 1$ 必然要选一个 $a_i = 1, b_i = 2$ 补上差距，对剩余需要的贡献是 3；$a_i = b_i$ 即该数直接贡献。

到这里我们简化了这个问题：有若干个 1，若干个 2，若干个 3，选出一些使得其和为某个数。

~~出题人不清楚这里二进制分组 $n \log n$ 能否通过，也没有特意去卡，望周知~~

此处给出线性构造。

分类讨论。

只有一种数很好办。

三种数都有，或者有 $1, 2$ 可以正负调整/两两分组，随便做，详见 P9498。

其余两种情况都有 3 和唯一的另外一种数，按模 3 讨论即可。

2、排序(sort)

比较套路的一道题吧。

首先是一个性质，对于任意一个子区间，翻转的区间互不相交。最大化不参与排序的元素个数即为答案。

考虑元素在什么区间里能不参与排序。很容易看出，左部分小于等于该元素，右部分大于等于该元素的区间就可以对答案进行 -1 的贡献。

对于 Sub2 ，树状数组+二分离线求得左右区间。对于 Sub3 ，用单调栈求得左右区间。

感谢 mcw 同学吊打出题人正解，加强了数据。

3、简单的选择题(mex)

题目背景的小 Z 是出题人(lg uid 306050)捏，羡慕小 H 和小 L。

很自然地想到对于每个二元组把它看成一条边建图。

然后对于建出来的图的连通块进行分类讨论：

对于一个 k 个点，t 条边的连通块：

首先给出一个结论：最优策略下连通块中只有至多一个数不能被选择奇数次。证明考虑假设最优方案有两个数 u, v 不能被选择，那么只需要对于 u, v 的一条路径上的所有边都取相反的一段就可以使 u, v 都满足，矛盾。

根据奇偶性可以得到，当 $t = k + 2a - 1, a \in \mathbb{N}$ 时，有一个点不能被选择，此时贪心选择标号最大的点；当 $t = k + 2a, a \in \mathbb{N}$ 时，所有的点都可以被选择奇数次。

那么至此只需要在线维护连通块的块内最大值和块内边数，显然可以使用并查集，每次修改只会对于最多两个点能否被取到进行更新。答案查询可以使用优先队列，出题人不确定有没有更好的查询方式。

时间复杂度 $O(n \log n + n\alpha(n))$。

4、亲合力(avidity)

Task 1：

暴力枚举，期望 10 分。

时间复杂度 $\mathcal{O}(mn^3)$。

Task 2：

用一个前缀和来维护每个数的个数，然后询问的时候就直接输出。期望得分 10 分。

Task 3：

先对查询区间排序，然后枚举 i 和 j，可以发现 k 单调不减。便可以二分 k 的值。

时间复杂度 $\mathcal{O}(mn^2 \log n)$。

或者直接用一个指针来代表 k，然后一直向后移看是否合法。

时间复杂度 $\mathcal{O}(mn^2)$。

期望得分 10 分。

实现得不好可能过不了 10^3 的数据。

Task 4：

这一档就有正解的思想了。

先预处理出所有的勾股数对，然后存起来。具体怎么处理见 Task 5。

然后查询的时候先开个桶记录，接着枚举 i，靠预处理的勾股数找是否存在满足条件的 j 和 k。

时间复杂度 $\mathcal{O}(kmn + a\sqrt{a})$。

Task 5：

留个不会处理勾股数的“正解”。

或者正解的大常数选手。

时间复杂度 $\mathcal{O}(k(m+n)\sqrt{n} + a^3)$

Task 6：

题目其实就是求区间 $[l, r]$ 的勾股数的对数。

求勾股数的方法：

先处理 a, b, c 互质的数，然后再乘一个常数。

因为 $a^2 + b^2 = c^2$，所以 $a^2 = (c+b)(c-b)$，设 $k = c - b$，则 $a^2 = k(2b+k)$，将 a 和 k 试做常数，解得 $b = \dfrac{a^2 - k^2}{2k}, c = \dfrac{a^2 + k^2}{2k}$。带入化简得到 $(2ka)^2 + (a^2 - k^2)^2 = (a^2 + k^2)^2$。

先处理出 a 中所有的勾股数。具体是以 a_i 为最小值时另外两个值时多少，以 a_i 为次大值时另外两个数时多少，以 a_i 为最大值时另外两个值时多少。

打表可以发现满足的另外两个值不会很多，当 $a_i \le 5 \times 10^4$ 时最多为 91 个。设这个的对数最多为 k。

然后分块。

预处理出每块中的勾股数对数，以及每块中每个位置到其块首和块尾到勾股数对数，时间复杂度 $\mathcal{O}(kn\sqrt{n})$。

然后处理出第 i 块到第 j 块的答案 $ans_{i,j}$，时间复杂度 $\mathcal{O}(kn\sqrt{n})$。

查询时需要分类讨论，分类 i, j, k 分别在的块。

令左边的散块为 `L`，中间的整块为 `M`，右边的散块为 `R`。

这个时候就要大力分类，分以下四大类，总计八小类：

1. `LLL`, `RRR`, `MMM`
2. `LLM`, `LMM`
3. `MRR`, `MMR`
4. `LMR`

第一大类可以由预处理直接得来。

第二大类可以枚举 L，然后统计得来。

第三大类可以枚举 R，然后统计得来。

第四大类可以枚举 L 或 R，然后统计得来。具体可以和第二或第三大类一起处理。

单词询问时间复杂度 $\mathcal{O}(k\sqrt{n})$。

总时间复杂度 $\mathcal{O}(k(m+n)\sqrt{n}+a\sqrt{a})$。

注意常数。

一个小优化，直接把没有出现过的数字直接删掉。