杂题选讲

九条可怜

杭州天水幼儿园

TCO2016 Round 2B easy TriangleTriples

给出 A, B, C，问满足 $a \leq A, b \leq B, c \leq C$ 的三角形个数，对一个大质数取模。

$A, B, C \leq 10^9$

TCO2016 Round 2B easy TriangleTriples

- 令 $f(A,B,C)$ 为满足 $x \le A, y \le B, z \le C, x+y \le z$ 的三元组 (x,y,z) 个数。

TCO2016 Round 2B easy TriangleTriples

- 令 $f(A,B,C)$ 为满足 $x \le A, y \le B, z \le C, x+y \le z$ 的三元组 (x,y,z) 个数。
- $ABC - f(A,B,C) - f(B,C,A) - f(A,C,B)$

TCO2016 Round 2B easy TriangleTriples

- 令 $f(A,B,C)$ 为满足 $x \le A, y \le B, z \le C, x+y \le z$ 的三元组 (x,y,z) 个数。
- $ABC - f(A,B,C) - f(B,C,A) - f(A,C,B)$
- 令 $g(A)$ 为 $z \le A, x+y \le z$ 的三元组个数。

- 令 $f(A, B, C)$ 为满足 $x \le A, y \le B, z \le C, x + y \le z$ 的三元组 (x, y, z) 个数。
- $ABC - f(A, B, C) - f(B, C, A) - f(A, C, B)$
- 令 $g(A)$ 为 $z \le A, x + y \le z$ 的三元组个数。
- $g(A) = \sum_{i=2}^{A}(i-1)(A-i+1)$。

TCO2016 Round 2B easy TriangleTriples

- 令 $f(A,B,C)$ 为满足 $x \le A, y \le B, z \le C, x+y \le z$ 的三元组 (x,y,z) 个数。
- $ABC - f(A,B,C) - f(B,C,A) - f(A,C,B)$
- 令 $g(A)$ 为 $z \le A, x+y \le z$ 的三元组个数。
- $g(A) = \sum_{i=2}^{A}(i-1)(A-i+1)$。
- $f(A,B,C) = g(C) - g(C-A) - g(C-B) + g(C-A-B)$。

- 令 $f(A,B,C)$ 为满足 $x \le A, y \le B, z \le C, x+y \le z$ 的三元组 (x,y,z) 个数。
- $ABC - f(A,B,C) - f(B,C,A) - f(A,C,B)$
- 令 $g(A)$ 为 $z \le A, x+y \le z$ 的三元组个数。
- $g(A) = \sum_{i=2}^{A}(i-1)(A-i+1)$。
- $f(A,B,C) = g(C) - g(C-A) - g(C-B) + g(C-A-B)$。
- 时间复杂度 $O(1)$。

有 n 种物品和 m 个背包，每一个背包里放了若干个物品（同一种可能有多个）。

问对于任意的物品单位重量的顺序，你是否总是可以确定最重的背包。

$n \leq 16, m \leq 50$

- 假设已经知道了物品重量的相对顺序，A 一定比 B 重当且仅当对于任意的 i，A 中属于前 i 重的种类的物品数量都不小于 B。

TCO2016 Round 2B medium FoxAndGemstone

- 假设已经知道了物品重量的相对顺序，A 一定比 B 重当且仅当对于任意的 i，A 中属于前 i 重的种类的物品数量都不小于 B。
- 令 $f(i)$ 为使第 i 个包最重的相对顺序数目。$\sum_{i=1}^{m} f(i) = n!$。

- 假设已经知道了物品重量的相对顺序，A 一定比 B 重当且仅当对于任意的 i，A 中属于前 i 重的种类的物品数量都不小于 B。
- 令 $f(i)$ 为使第 i 个包最重的相对顺序数目。$\sum_{i=1}^{m} f(i) = n!$。
- 状压 DP。

TCO2016 Round 2B medium FoxAndGemstone

- 假设已经知道了物品重量的相对顺序，A 一定比 B 重当且仅当对于任意的 i，A 中属于前 i 重的种类的物品数量都不小于 B。
- 令 $f(i)$ 为使第 i 个包最重的相对顺序数目。$\sum_{i=1}^{m} f(i) = n!$。
- 状压 DP。
- 时间复杂度 $O(2^n nm)$

对一个 1 到 n 的排列 A，你可以进行若干次操作，每次操作选出一个满足 $i < j$ 且 $A_i > A_j$ 的数对 (i, j) 并交换 A_i, A_j。

如果排列 B 能由 A 进行若干次变化得到，那么 B 被称为可到达的。

给出 A，问可到达的排列数目，对一个大质数取模。

$n \leq 20$

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。
- 可以将排列 A 转化成 n 个 01 序列 a_i，$a_{i,j} = (A_i \geq j)$。

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。
- 可以将排列 A 转化成 n 个 01 序列 a_i，$a_{i,j} = (A_i \geq j)$。
- 排列 A 能到达 B 的必要条件是 a_i 能到达 b_i。

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。
- 可以将排列 A 转化成 n 个 01 序列 a_i，$a_{i,j} = (A_i \geq j)$。
- 排列 A 能到达 B 的必要条件是 a_i 能到达 b_i。
- 充分性：从 n 到 1 枚举 i，要把 i 从 l 移动到 r，每一次找到 $(l, r]$ 中最大的位置 x，并将 l 和 x 交换。

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。
- 可以将排列 A 转化成 n 个 01 序列 a_i，$a_{i,j} = (A_i \geq j)$。
- 排列 A 能到达 B 的必要条件是 a_i 能到达 b_i。
- 充分性：从 n 到 1 枚举 i，要把 i 从 l 移动到 r，每一次找到 $(l, r]$ 中最大的位置 x，并将 l 和 x 交换。
- 状压 DP。

hihocoder 挑战赛 21 Reachable Permutations

- 对于 01 序列来说，A 能到达 B 当且仅当对于每一个 i，B 中第 i 个 1 都不在 A 的左边。
- 可以将排列 A 转化成 n 个 01 序列 a_i，$a_{i,j} = (A_i \geq j)$。
- 排列 A 能到达 B 的必要条件是 a_i 能到达 b_i。
- 充分性：从 n 到 1 枚举 i，要把 i 从 l 移动到 r，每一次找到 $(l, r]$ 中最大的位置 x，并将 l 和 x 交换。
- 状压 DP。
- 时间复杂度 $O(2^n n)$。

有 n 个工作，n 是偶数，每一个工作都有一个经验值 a_i 和报酬 b_i。你最开始经验值 E 是 0。

每做完一个工作，首先你的经验值会增加 a_i，接着你会收到 $E \times b_i$ 的报酬。

在完成了 $\frac{n}{2}$ 个工作后你的经验值会额外增加 X。

现在你需要安排完成这 n 个工作的顺序，使得在做完每一个工作后，得到的报酬之和最大。

$n \leq 50, X, a_i \leq 10^5, b_i \leq 10$

SRM691 div1-medium Moneymanager

- 前半段和后半段一定是按照 $\frac{b_i}{a_i}$ 升序完成。

SRM691 div1-medium Moneymanager

- 前半段和后半段一定是按照 $\frac{b_i}{a_i}$ 升序完成。
- 排序后令 $f[i][j][k]$ 表示当前考虑到第 i 个，前半段选了 j 个，选取的后半段的 b_i 和为 k 时收益的最大值。

- 前半段和后半段一定是按照 $\frac{b_i}{a_i}$ 升序完成。
- 排序后令 $f[i][j][k]$ 表示当前考虑到第 i 个，前半段选了 j 个，选取的后半段的 b_i 和为 k 时收益的最大值。
- 无法统计贡献，需要知道后半段所有工作的 b_i 的和才能转移。

SRM691 div1-medium Moneymanager

- 前半段和后半段一定是按照 $\frac{b_i}{a_i}$ 升序完成。
- 排序后令 $f[i][j][k]$ 表示当前考虑到第 i 个，前半段选了 j 个，选取的后半段的 b_i 和为 k 时收益的最大值。
- 无法统计贡献，需要知道后半段所有工作的 b_i 的和才能转移。
- 枚举后半段所有工作的 b_i 的和，对每一个可能的和都 DP 一遍。

- 前半段和后半段一定是按照 $\frac{b_i}{a_i}$ 升序完成。
- 排序后令 $f[i][j][k]$ 表示当前考虑到第 i 个，前半段选了 j 个，选取的后半段的 b_i 和为 k 时收益的最大值。
- 无法统计贡献，需要知道后半段所有工作的 b_i 的和才能转移。
- 枚举后半段所有工作的 b_i 的和，对每一个可能的和都 DP 一遍。
- $O(n^2(\sum b_i)^2)$

给出 n 个容器，每个容器有参数 mi_i, ma_i, c_i, p_i。同时有 m 种液体，每种液体有 v_i 毫升。

给出一种方案使得所有液体都被装进了容器里，并满足每一个容器里液体的体积之和在 $[mi_i, ma_i]$ 中，且第 i 个容器中第 c_i 种液体的比例不少于 $\frac{p_i}{100}$。可能无解。

$n, m \leq 10^5$。

- 先考虑限制部分，当可以加入的体积等于液体总体积时，再把剩下的液体加进去。

- 先考虑限制部分，当可以加入的体积等于液体总体积时，再把剩下的液体加进去。
- 先满足每一个容器 mi_i 的限制，然后贪心的增加总体积。

- 先考虑限制部分，当可以加入的体积等于液体总体积时，再把剩下的液体加进去。
- 先满足每一个容器 mi_i 的限制，然后贪心的增加总体积。
- 按照 $\frac{100-p_i}{p_i}$ 排序贪心的加限制部分。

- 先考虑限制部分，当可以加入的体积等于液体总体积时，再把剩下的液体加进去。
- 先满足每一个容器 mi_i 的限制，然后贪心的增加总体积。
- 按照 $\frac{100-p_i}{p_i}$ 排序贪心的加限制部分。
- 时间复杂度 $O(n \log n)$

一个 idea

给出 k，解方程 $\sum_{i=1}^{x} \mu(i) = k$。

如果在 $[1, 10^9]$ 内无解输出 -1，否则输出任意一个解。

一个 idea

- 令 $f(x)=\sum_{i=1}^{x}\mu(x)$，性质 $|f(x)-f(x-1)|\le 1$。

一个 idea

- 令 $f(x)=\sum_{i=1}^{x}\mu(x)$，性质 $|f(x)-f(x-1)|\le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。

一个 idea

- 令 $f(x)=\sum_{i=1}^{x}\mu(x)$，性质 $|f(x)-f(x-1)|\le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。
- 二分，初始区间为 $[\min(\mathsf{mi},\mathsf{ma}),\max(\mathsf{mi},\mathsf{ma})]$。

一个 idea

- 令 $f(x) = \sum_{i=1}^{x} \mu(x)$，性质 $|f(x) - f(x-1)| \le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。
- 二分，初始区间为 $[\min(\mathsf{mi}, \mathsf{ma}), \max(\mathsf{mi}, \mathsf{ma})]$。
- 若 $f(\mathsf{mid}) < k$，划向最大值那边，否则划向最小值那边。

一个 idea

- 令 $f(x)=\sum_{i=1}^{x}\mu(x)$，性质 $|f(x)-f(x-1)|\le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。
- 二分，初始区间为 $[\min(\mathsf{mi},\mathsf{ma}),\max(\mathsf{mi},\mathsf{ma})]$。
- 若 $f(\mathsf{mid})<k$，划向最大值那边，否则划向最小值那边。
- $f(n)$ 用杜教筛求，单次 $O(n^{\frac{2}{3}})$。

一个 idea

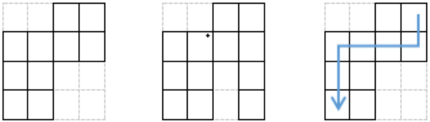
- 令 $f(x) = \sum_{i=1}^{x} \mu(x)$，性质 $|f(x) - f(x-1)| \le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。
- 二分，初始区间为 $[\min(\mathsf{mi}, \mathsf{ma}), \max(\mathsf{mi}, \mathsf{ma})]$。
- 若 $f(\mathsf{mid}) < k$，划向最大值那边，否则划向最小值那边。
- $f(n)$ 用杜教筛求，单次 $O(n^{\frac{2}{3}})$。
- 分段筛求最大值与最小值，可以做到 10^9 内所有数的约数个数和。

一个 idea

- 令 $f(x)=\sum_{i=1}^{x}\mu(x)$，性质 $|f(x)-f(x-1)|\le 1$。
- 预处理出 $f(x)$ 的最大值位置 ma 与最小值位置 mi。
- 二分，初始区间为 $[\min(\mathsf{mi},\mathsf{ma}),\max(\mathsf{mi},\mathsf{ma})]$。
- 若 $f(\mathsf{mid})<k$，划向最大值那边，否则划向最小值那边。
- $f(n)$ 用杜教筛求，单次 $O(n^{\frac{2}{3}})$。
- 分段筛求最大值与最小值，可以做到 10^9 内所有数的约数个数和。
- 时间复杂度 $O(n^{\frac{2}{3}}\log n)$

对于一个凸的四联通块，定义一条路径的权值为转弯次数，两点之间的距离为路径权值最小值，直接定义为距离最大的点对之间的距离。

给出一个凸的四联通块，求其直径。



$n, m \leq 2000$

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。
- 令 $R[i][j]$ 为 (i,j) 向右走能走到的最后一个点。

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。
- 令 $R[i][j]$ 为 (i,j) 向右走能走到的最后一个点。
- 令 $D[i][j]$ 为 (i,j) 向下走能走到的最后一个点。

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。
- 令 $R[i][j]$ 为 (i,j) 向右走能走到的最后一个点。
- 令 $D[i][j]$ 为 (i,j) 向下走能走到的最后一个点。
- 横坐标 $x \in [i, R[i][j]]$ 或纵坐标 $y \in [j, D[i][j]]$ 的点可以一步走到。

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。
- 令 $R[i][j]$ 为 (i,j) 向右走能走到的最后一个点。
- 令 $D[i][j]$ 为 (i,j) 向下走能走到的最后一个点。
- 横坐标 $x \in [i, R[i][j]]$ 或纵坐标 $y \in [j, D[i][j]]$ 的点可以一步走到。
- $f[i][j] = f[R[i][j]][D[i][j]] + 1$。

ASC 47 Jinxiety of a Polyomino

- 令 $f[i][j]$ 为 (i,j) 到其右下方所有点距离的最大值。
- 令 $R[i][j]$ 为 (i,j) 向右走能走到的最后一个点。
- 令 $D[i][j]$ 为 (i,j) 向下走能走到的最后一个点。
- 横坐标 $x \in [i, R[i][j]]$ 或纵坐标 $y \in [j, D[i][j]]$ 的点可以一步走到。
- $f[i][j] = f[R[i][j]][D[i][j]] + 1$。
- $O(nm)$

某 WC 模拟题 oddpaths

给出一张 n 个点的有向无环图，每一个点都有一个颜色，其中有 K 个点可能是黑的可能是白的，剩下的点都是白的。总共有 2^K 种情况，问其中有多少种满足 1 号点到 n 号点只经过白色点的路径条数为奇数。

$n \leq 50, K \leq 32$

某 WC 模拟题 oddpaths

- 令 $w[i][j]$ 为 i 到 j 的路径条数，$f[i][j]$ 为 i 到 j 不经过拓扑序小于 j 的黑点的路径条数。

某 WC 模拟题 oddpaths

- 令 $w[i][j]$ 为 i 到 j 的路径条数，$f[i][j]$ 为 i 到 j 不经过拓扑序小于 j 的黑点的路径条数。
- 如果知道了哪些是黑点，$O(K^2)$ DP。

某 WC 模拟题 oddpaths

- 令 $w[i][j]$ 为 i 到 j 的路径条数，$f[i][j]$ 为 i 到 j 不经过拓扑序小于 j 的黑点的路径条数。
- 如果知道了哪些是黑点，$O(K^2)$ DP。
- 分两段状压 DP，前半段的状态是前 $\frac{K}{2}$ 的待定点选了哪些，后半段的状态是对后 $\frac{K}{2}$ 的每一个待定点 i，记 $f[1][i]$ 的奇偶性。

某 WC 模拟题 oddpaths

- 令 $w[i][j]$ 为 i 到 j 的路径条数，$f[i][j]$ 为 i 到 j 不经过拓扑序小于 j 的黑点的路径条数。
- 如果知道了哪些是黑点，$O(K^2)$ DP。
- 分两段状压 DP，前半段的状态是前 $\frac{K}{2}$ 的待定点选了哪些，后半段的状态是对后 $\frac{K}{2}$ 的每一个待定点 i，记 $f[1][i]$ 的奇偶性。
- DP 完前 $\frac{K}{2}$ 个点后，$O(2^{\frac{K}{2}})$ 转换状态。

某 WC 模拟题 oddpaths

- 令 $w[i][j]$ 为 i 到 j 的路径条数，$f[i][j]$ 为 i 到 j 不经过拓扑序小于 j 的黑点的路径条数。
- 如果知道了哪些是黑点，$O(K^2)$ DP。
- 分两段状压 DP，前半段的状态是前 $\frac{K}{2}$ 的待定点选了哪些，后半段的状态是对后 $\frac{K}{2}$ 的每一个待定点 i，记 $f[1][i]$ 的奇偶性。
- DP 完前 $\frac{K}{2}$ 个点后，$O(2^{\frac{K}{2}})$ 转换状态。
- 时间复杂度 $O(2^{\frac{K}{2}})$。

GCJ2016 Round 3 D Go++

语言 go++ 包含三种指令 $0, 1, ?$ 和一个 01 变量 x，初始 x 为 0。

0 会令 $x = 0$1 会令 $x = 1$? 会输出 x，例如 ?0?01? 会输出 001。

定义两个 go++ 代码的组合为它们相互穿插形成的代码集，例如 1? 和 $\underline{?0}$ 的组合为 $\{\underline{?0}1?, \underline{?}1\underline{0}?, \underline{?}1?\underline{0}, 1\underline{?0}?, 1?\underline{?0}, 1\underline{?}?\underline{0}\}$。

现在给出 n 个长度为 L 的好的串以及一个长度为 L 的坏的串。你需要构造两个长度不超过 200 的 go++ 代码，使得对于任意一个好的串，它都至少是一个组合集中的代码的输出；对于坏的串，它不是组合集中任意一个代码的输出。无解输出 IMPOSSIBLE。

$n \leq 100, L \leq 50$

GCJ2016 Round 3 D Go++

- 设不好的串是 s。

GCJ2016 Round 3 D Go++

- 设不好的串是 s。
- $(!s_1)?(!s_2)?...(!s_n)?$

- 设不好的串是 s。
- $(!s_1)?(!s_2)?...(!s_n)?$
- $(!s_1)s_1(!s_2)s_2...(!s_{n-1})s_{n-1}$

Snackdown2016 Pre-elimination round A MAKELIS

给出一个 K，你需要给出 n 和一个长度为 n 的排列 A，使得 A 中不同的最长上升子序列个数恰好为 K。n 不能超过 100。

$(1,3,2,4)$ 有两个不同的最长上升子序列 $(1,3,4)$ 和 $(1,2,4)$。

$K \leq 10^5$

- $\times 2$：在最后加上 $\infty, \infty - 1$，例如 $(3,1,2)$ 变成 $(3,1,2,5,4)$。

- $\times 2$：在最后加上 $\infty, \infty - 1$，例如 $(3, 1, 2)$ 变成 $(3, 1, 2, 5, 4)$。
- $+1$：设当前 LIS 程度为 n，在最后加上 $-\infty, ..., -\infty + n - 1$，例如 $(3, 1, 2)$ 变成 $(5, 3, 4, 1, 2)$。

- $\times 2$：在最后加上 $\infty, \infty - 1$，例如 $(3,1,2)$ 变成 $(3,1,2,5,4)$。
- $+1$：设当前 LIS 程度为 n，在最后加上 $-\infty, ..., -\infty + n - 1$，例如 $(3,1,2)$ 变成 $(5,3,4,1,2)$。
- $O(\log^2 K)$。

Snackdown2016 Pre-elimination round A MAKELIS

- 要充分利用最长上升子序列的最长的性质。

- 要充分利用最长上升子序列的最长的性质。
- 考虑这样的序列 $(1, \infty|3, 2, \infty+1|...|2n+1, 2n, \infty+n|2n+2)$。

- 要充分利用最长上升子序列的最长的性质。
- 考虑这样的序列 $(1, \infty | 3, 2, \infty + 1 | ... | 2n + 1, 2n, \infty + n | 2n + 2)$。
- 最长上升子序列长度为 $n + 2$，个数为 2^n，一旦选了高一层的数字，最多长度最多只能达到 $n + 1$。其中，通过 $\infty + i$ 接入的这样的上升序列个数为 2^i。

- 要充分利用最长上升子序列的最长的性质。
- 考虑这样的序列 $(1, \infty | 3, 2, \infty + 1 | ... | 2n + 1, 2n, \infty + n | 2n + 2)$。
- 最长上升子序列长度为 $n + 2$，个数为 2^n，一旦选了高一层的数字，最多长度最多只能达到 $n + 1$。其中，通过 $\infty + i$ 接入的这样的上升序列个数为 2^i。
- 用 $\infty - i - 1$ 来将 2^i 加入答案中。

- 要充分利用最长上升子序列的最长的性质。
- 考虑这样的序列 $(1,\infty|3,2,\infty+1|...|2n+1,2n,\infty+n|2n+2)$。
- 最长上升子序列长度为 $n+2$，个数为 2^n，一旦选了高一层的数字，最多长度最多只能达到 $n+1$。其中，通过 $\infty+i$ 接入的这样的上升序列个数为 2^i。
- 用 $\infty-i-1$ 来将 2^i 加入答案中。
- 例如需要构造的数是 2^n+3，那么就构造数列：

$$(1,\infty-1,\infty|3,2,\infty-2,\infty+1|...|2n+1,2n,\infty+n|2n+2)$$

- 要充分利用最长上升子序列的最长的性质。
- 考虑这样的序列 $(1, \infty | 3, 2, \infty + 1 | ... | 2n + 1, 2n, \infty + n | 2n + 2)$。
- 最长上升子序列长度为 $n + 2$，个数为 2^n，一旦选了高一层的数字，最多长度最多只能达到 $n + 1$。其中，通过 $\infty + i$ 接入的这样的上升序列个数为 2^i。
- 用 $\infty - i - 1$ 来将 2^i 加入答案中。
- 例如需要构造的数是 $2^n + 3$，那么就构造数列：

$$(1, \infty - 1, \infty | 3, 2, \infty - 2, \infty + 1 | ... | 2n + 1, 2n, \infty + n | 2n + 2)$$

- $O(\log K)$。