# 模拟赛讲评

Vxlimo

福建师范大学附属中学

2023 年 7 月 10 日

# 图 (graph)

# 图 (graph)

给你一个 $n$ 个点，$m$ 条边的无向连通有权图，将其中任意不同两点的距离从小到大排列。

求这个排列中第 $k$ 个元素的大小，也即图中第 $k$ 短的最短路。

# 吐槽环节

# 解法 1

# 解法 1

- $n \leq 1000$。

# 解法 1

- $n \leq 1000$。
- 直接 floyd 然后排序，时间复杂度 $\mathcal{O}(n^3)$，可以拿到 $30Pts$ 的好成绩。

# 解法 2

# 解法 2

- $n \le 2 \times 10^5, k \le 400$。

# 解法 2

- $n \le 2 \times 10^5, k \le 400$。
- $k$ 很小，我们考虑从 $k$ 入手。

# 解法 2

- $n \leq 2 \times 10^5, k \leq 400$。
- $k$ 很小，我们考虑从 $k$ 入手。
- 注意到我们可以直接选前 $k$ 小的边，最后的答案一定由这些边的集合构成。

# 解法 2

- $n \le 2 \times 10^5, k \le 400$。
- $k$ 很小，我们考虑从 $k$ 入手。
- 注意到我们可以直接选前 $k$ 小的边，最后的答案一定由这些边的集合构成。
- 这是因为没有负边，加边一定会导致长度增加。

# 解法 2

- $n \le 2 \times 10^5, k \le 400$。
- $k$ 很小，我们考虑从 $k$ 入手。
- 注意到我们可以直接选前 $k$ 小的边，最后的答案一定由这些边的集合构成。
- 这是因为没有负边，加边一定会导致长度增加。
- 因此点的个数减少到 $2 \times k$ 个，时间复杂度 $\mathcal{O}(k^3)$，可以得到 $100Pts$ 的好成绩。

# 战争（war）

# 战争 (war)

现在有一片大小为 $p \times p$ 的战场，国王想要把他手下的 $m$ 种单位全部派到这片战场上。

我们称一个摆放方法为一个“战斗阵型”，当且仅当对任意一个 $2 \times 2$ 的子矩阵，都有如下两个性质：

- $4$ 个位置中最多有 $3$ 个摆放了单位。
- 一条对角线上的两个位置如果都有摆放单位，那么他们不能是同一种单位。

现在国王给你每种单位的数量，他想知道满足条件的最小的“战斗阵型”的边长，并希望你给出对应的方案。

# 吐槽环节

# 解法 1

# 解法 1

- 没有重复单位。

# 解法 1

- 没有重复单位。
- 题目限制要求每 $4$ 个放最多 $3$ 个，不难想到模块化构造 L 型，可以满足恰好放 $3$ 个，这样在满足条件的情况下，阵型一定是最小的。

# 解法 1

- 没有重复单位。
- 题目限制要求每 $4$ 个放最多 $3$ 个，不难想到模块化构造 L 型，可以满足恰好放 $3$ 个，这样在满足条件的情况下，阵型一定是最小的。
- 即阵型面积约为 $\frac{4}{3}m$ （要再小一点，阵型边长为奇数时下面多出来一整行，右边多出来一整列）。

# 解法 1

- 没有重复单位。
- 题目限制要求每 $4$ 个放最多 $3$ 个，不难想到模块化构造 L 型，可以满足恰好放 $3$ 个，这样在满足条件的情况下，阵型一定是最小的。
- 即阵型面积约为 $\frac{4}{3}m$ （要再小一点，阵型边长为奇数时下面多出来一整行，右边多出来一整列）。
- 时间复杂度 $O(\sum n+\sum m)$，可以得到 $20Pts$ 的好成绩。

# 解法 2

# 解法 2

- 最多两种单位。

# 解法 2

- 最多两种单位。
- 与上一档部分分不同的是，第二种限制开始发挥作用。继续考虑 L 型?

# 解法 2

- 最多两种单位。
- 与上一档部分分不同的是，第二种限制开始发挥作用。继续考虑 L 型？
- 首先发现，空出来的那个格子的对角格子可以随便放，而另外两个格子要求不能放同一种单位。

# 解法 2

- 最多两种单位。
- 与上一档部分分不同的是，第二种限制开始发挥作用。继续考虑 L 型？
- 首先发现，空出来的那个格子的对角格子可以随便放，而另外两个格子要求不能放同一种单位。
- 然后，对于出现次数较多的单位，显然最多也只能把对角线的一个格子和随便填的格子放满。

# 解法 2

# 解法 2

- 尝试满足阵型最小的条件，令出现较多次的单位出现了 $mx$ 次，容易发现阵型面积的下界约为 $\frac{4}{3}m$ 和 $2mx$ （同样再小一点，阵型边长为奇数时下面多出来一整行，右边也是再多一整列）的较小值。

# 解法 2

- 尝试满足阵型最小的条件，令出现较多次的单位出现了 $mx$ 次，容易发现阵型面积的下界约为 $\frac{4}{3}m$ 和 $2mx$ （同样再小一点，阵型边长为奇数时下面多出来一整行，右边也是再多一整列）的较小值。
- 也即阵型边长 $p$ 满足 $p^2 - \lfloor \frac{p}{2} \rfloor^2 \geq m, n \cdot \lfloor \frac{(n+1)}{2} \rfloor \geq mx$。

# 解法 2

- 尝试满足阵型最小的条件，令出现较多次的单位出现了 $mx$ 次，容易发现阵型面积的下界约为 $\frac{4}{3}m$ 和 $2mx$ （同样再小一点，阵型边长为奇数时下面多出来一整行，右边也是再多一整列）的较小值。
- 也即阵型边长 $p$ 满足 $p^2 - \lfloor \frac{p}{2} \rfloor^2 \geq m, n \cdot \lfloor \frac{(n+1)}{2} \rfloor \geq mx$。
- 当阵型面积恰为这个下界时能否满足条件?

# 解法 2

# 解法 2

- 我们把出现次数较多的那个单位填进对角线的一个格子和随便填的格子，由于下界的存在一定足够填。

# 解法 2

- 我们把出现次数较多的那个单位填进对角线的一个格子和随便填的格子，由于下界的存在一定足够填。
- 将另一个单位填进剩下的对角线的这个格子，对角线的另一个格子和随便填的格子，同样由于下界的存在一定足够填，并且由于它的出现次数较少，也不会冲突。

# 解法 2

- 我们把出现次数较多的那个单位填进对角线的一个格子和随便填的格子，由于下界的存在一定足够填。
- 将另一个单位填进剩下的对角线的这个格子，对角线的另一个格子和随便填的格子，同样由于下界的存在一定足够填，并且由于它的出现次数较少，也不会冲突。
- 时间复杂度 $O(\sum n+\sum m)$，结合解法 $1$ 可以得到 $60Pts$ 的好成绩。

# 解法 3

# 解法 3

- 都会 $m=2$ 了不会一般情况?

# 解法 3

- 都会 $m=2$ 了不会一般情况?
- 同样地，找到出现次数最多的格子，按照解法 2 的方法一路填就好了。

# 解法 3

- 都会 $m = 2$ 了不会一般情况?
- 同样地，找到出现次数最多的格子，按照解法 2 的方法一路填就好了。
- 时间复杂度 $O(\sum n + \sum m)$，可以得到 $100Pts$ 的好成绩。

# 树 (tree)

# 树 (tree)

我们称一棵权值为 $n$ 的树是“好的”，当且仅当：

- 它的若干个（至少 $2$ 个）子树均为权值为 $k$，结构完全一样的“好的”树，且权值和不超过 $n$。

- 称“浪费的权值”为 $n-$ 子树权值和，浪费的权值和 $< k$。

特别地，一个点被视作权值为 $1$ 的“好的”树。

你需要求出权值为 $n$ 的，“好的”树的形态数量。

# 吐槽环节

# 解法 1

# 解法 1

- $n \leq 1000$。

# 解法 1

- $n \leq 1000$。
- 考虑 dp，设 $f_i$ 表示权值为 $i$ 的“好的”树的个数。直接按照题目的要求，枚举子树的个数和权值转移即可。

# 解法 1

- $n \le 1000$。
- 考虑 dp，设 $f_i$ 表示权值为 $i$ 的“好的”树的个数。直接按照题目的要求，枚举子树的个数和权值转移即可。
- $f_i = \sum_{j=2}^{i} \sum_{k=1}^{i} f_k$。其中，$j \times k \le i, i - j \times k < k$。

# 解法 1

- $n \leq 1000$。
- 考虑 dp，设 $f_i$ 表示权值为 $i$ 的“好的”树的个数。直接按照题目的要求，枚举子树的个数和权值转移即可。
- $f_i = \sum_{j=2}^{i} \sum_{k=1}^{i} f_k$。其中，$j \times k \leq i, i - j \times k < k$。
- 时间复杂度 $O(n^3)$，常数应该比较小，可以得到 $20Pts$ 的好成绩。

# 解法 2

# 解法 2

- $n \leq 5 \times 10^4$。

# 解法 2

- $n \le 5 \times 10^4$。
- 考虑权值为 $w$ 的树，有 $k$ 个权值为 $w'$ 的子树。那么有 $w = k \times w' + p$，且 $p < k$。

# 解法 2

- $n \le 5 \times 10^4$。
- 考虑权值为 $w$ 的树，有 $k$ 个权值为 $w'$ 的子树。那么有 $w = k \times w' + p$，且 $p < k$。
- 容易发现，如果 $k$ 确定，$w'$ 实际上只有一种取值，$w' = \lfloor \frac{w}{k} \rfloor$。

# 解法 2

- $n \leq 5 \times 10^4$。
- 考虑权值为 $w$ 的树，有 $k$ 个权值为 $w'$ 的子树。那么有 $w = k \times w' + p$，且 $p < k$。
- 容易发现，如果 $k$ 确定，$w'$ 实际上只有一种取值，$w' = \lfloor \frac{w}{k} \rfloor$。
- 或者你也可以由解法 1 的两个不等式，得到 $j$ 确定时，$k$ 只有一种取值合法。

# 解法 2

- $n \leq 5 \times 10^4$。
- 考虑权值为 $w$ 的树，有 $k$ 个权值为 $w'$ 的子树。那么有 $w = k \times w' + p$，且 $p < k$。
- 容易发现，如果 $k$ 确定，$w'$ 实际上只有一种取值，$w' = \lfloor \frac{w}{k} \rfloor$。
- 或者你也可以由解法 1 的两个不等式，得到 $j$ 确定时，$k$ 只有一种取值合法。
- 因此：$f_i = \sum_{j=2}^{i} f_{\lfloor \frac{i}{j} \rfloor}$。

# 解法 2

- $n \le 5 \times 10^4$。
- 考虑权值为 $w$ 的树，有 $k$ 个权值为 $w'$ 的子树。那么有 $w = k \times w' + p$，且 $p < k$。
- 容易发现，如果 $k$ 确定，$w'$ 实际上只有一种取值，$w' = \lfloor \frac{w}{k} \rfloor$。
- 或者你也可以由解法 1 的两个不等式，得到 $j$ 确定时，$k$ 只有一种取值合法。
- 因此：$f_i = \sum_{j=2}^{i} f_{\lfloor \frac{i}{j} \rfloor}$。
- 时间复杂度 $O(n^2)$，可以得到 $40Pts$ 的好成绩。

# 解法 3

# 解法 3

- $n \le 10^6$。

# 解法 3

- $n \leq 10^6$。
- 直接把解法 2 的式子用整除分块优化即可。

# 解法 3

- $n \leq 10^6$。
- 直接把解法 2 的式子用整除分块优化即可。
- 时间复杂度 $O\ n\sqrt{n})$，可以得到 $60Pts$ 的好成绩。

# 解法 4

# 解法 4

- $n \leq 10^9$。

# 解法 4

- $n \leq 10^9$。
- 注意到我们只需要知道 $f_n$ 的值而不需要求出所有的值。

# 解法 4

- $n \leq 10^9$。
- 注意到我们只需要知道 $f_n$ 的值而不需要求出所有的值。
- 记忆化搜索即可。

# 解法 4

- $n \le 10^9$。
- 注意到我们只需要知道 $f_n$ 的值而不需要求出所有的值。
- 记忆化搜索即可。
- 时间复杂度 $O(n^{\frac{3}{4}})$（证明类似杜教筛），可以得到 $100Pts$ 的好成绩。

# 序列 (sequence)

# 序列 (sequence)

有一个长度为 $n$ 的整数序列，你可以删除前面若干数和后面若干数（可以是 $0$ 个)。

求使剩下的序列中，某一个数出现的次数严格超过一半的方案数。

空序列不满足要求。

# 吐槽环节

# 解法 1

# 解法 1

- $n \leq 300$。

## 解法 1

- $n \leq 300$。
- 离散化后枚举区间，开个桶查询即可，时间复杂度 $O(n^3)$。

# 解法 1

- $n \le 300$。
- 离散化后枚举区间，开个桶查询即可，时间复杂度 $O(n^3)$。
- 可以得到 $5Pts$ 的好成绩。

# 解法 2

# 解法 2

- $n \leq 2000$。

# 解法 2

- $n \leq 2000$。
- 固定左端点，然后直接移动右端点的同时统计答案即可，时间复杂度 $O(n^2)$。

# 解法 2

- $n \leq 2000$。
- 固定左端点，然后直接移动右端点的同时统计答案即可，时间复杂度 $O(n^2)$。
- 可以得到 $15Pts$ 的好成绩。

# 解法 3

# 解法 3

- $\forall 1 \le i \le n, 1 \le val_i \le 2$。

# 解法 3

- $\forall 1 \le i \le n, 1 \le val_i \le 2$。
- 数只有两种，把一种当做 $-1$，另一种当做 $1$，这样问题转化为求区间和 $> 0$ 的区间个数。

# 解法 3

- $\forall 1 \le i \le n, 1 \le val_i \le 2$。
- 数只有两种，把一种当做 $-1$，另一种当做 $1$，这样问题转化为求区间和 $> 0$ 的区间个数。
- 再做一个前缀和，枚举右端点，问题转化为求有多少个数比它小。

# 解法 3

- $\forall 1 \le i \le n, 1 \le val_i \le 2$。
- 数只有两种，把一种当做 $-1$，另一种当做 $1$，这样问题转化为求区间和 $> 0$ 的区间个数。
- 再做一个前缀和，枚举右端点，问题转化为求有多少个数比它小。
- 树状数组或者权值线段树或者 $set$ 维护所有的左端点即可，时间复杂度 $O(n \log n)$。

# 解法 3

- $\forall 1 \leq i \leq n, 1 \leq val_i \leq 2$。
- 数只有两种，把一种当做 $-1$，另一种当做 $1$，这样问题转化为求区间和 $> 0$ 的区间个数。
- 再做一个前缀和，枚举右端点，问题转化为求有多少个数比它小。
- 树状数组或者权值线段树或者 $set$ 维护所有的左端点即可，时间复杂度 $O(n \log n)$。
- 结合解法 1 和解法 2，可以得到 $40Pts$ 的好成绩。

# 解法 4

# 解法 4

- $1 \le n \le 2 \times 10^5$。

## 解法 4

- $1 \leq n \leq 2 \times 10^5$。
- 发现解法 3 的做法可以推广到一般情况，对每种数，把它当做 $1$，剩下当做 $-1$，就和上面那个问题一样了。

# 解法 4

- $1 \le n \le 2 \times 10^5$。
- 发现解法 3 的做法可以推广到一般情况，对每种数，把它当做 1，剩下当做 $-1$，就和上面那个问题一样了。
- 但是这样的复杂度是 $O(nm)$ 的，其中 $m$ 是不同的数的个数，最大可以到 $n$，寄。

# 解法 4

- $1 \le n \le 2 \times 10^5$。
- 发现解法 3 的做法可以推广到一般情况，对每种数，把它当做 $1$，剩下当做 $-1$，就和上面那个问题一样了。
- 但是这样的复杂度是 $O(nm)$ 的，其中 $m$ 是不同的数的个数，最大可以到 $n$，寄。
- 如果要让复杂度正确，我们必须保证只遍历 $1$ 的位置，$-1$ 和 $-1$，$-1$ 和 $1$ 的贡献需要快速计算。

# 解法 4

- $1 \leq n \leq 2 \times 10^5$。
- 发现解法 3 的做法可以推广到一般情况，对每种数，把它当做 $1$，剩下当做 $-1$，就和上面那个问题一样了。
- 但是这样的复杂度是 $O(nm)$ 的，其中 $m$ 是不同的数的个数，最大可以到 $n$，寄。
- 如果要让复杂度正确，我们必须保证只遍历 $1$ 的位置，$-1$ 和 $-1$，$-1$ 和 $1$ 的贡献需要快速计算。
- 两段 $1$ 之间相当于权值线段树上区间加，这样 $-1$ 和 $1$ 的贡献就和 $1$ 和 $1$ 的贡献一样计算就好了。

# 解法 4

# 解法 4

- 不妨令两个 $1$ 的位置之间的前缀和数组为 $x+k, x+k-1, \cdots, x+1, x$，我们惊奇地发现，这一段 $-1$ 内部互相之间是没有贡献的，尝试直接算出一整段的贡献。

# 解法 4

- 不妨令两个 $1$ 的位置之间的前缀和数组为 $x+k, x+k-1, \cdots, x+1, x$，我们惊奇地发现，这一段 $-1$ 内部互相之间是没有贡献的，尝试直接算出一整段的贡献。
- 左端点为 $x+k-1$ 的算 $1$ 次，$x+k-2$ 的算 $2$ 次，以此类推直到 $\leq x-1$ 的剩下部分都是算 $k$ 次。

# 解法 4

- 不妨令两个 1 的位置之间的前缀和数组为 $x+k, x+k-1, \cdots, x+1, x$，我们惊奇地发现，这一段 $-1$ 内部互相之间是没有贡献的，尝试直接算出一整段的贡献。
- 左端点为 $x+k-1$ 的算 1 次，$x+k-2$ 的算 2 次，以此类推直到 $\leq x-1$ 的剩下部分都是算 $k$ 次。
- 这是经典问题，使用线段树维护区间和的同时，维护 $\sum_{i=1}^{len}(len-i+1)val_i$ 即可。

# 解法 4

- 不妨令两个 1 的位置之间的前缀和数组为 $x+k, x+k-1, \cdots, x+1, x$，我们惊奇地发现，这一段 $-1$ 内部互相之间是没有贡献的，尝试直接算出一整段的贡献。
- 左端点为 $x+k-1$ 的算 1 次，$x+k-2$ 的算 2 次，以此类推直到 $\le x-1$ 的剩下部分都是算 $k$ 次。
- 这是经典问题，使用线段树维护区间和的同时，维护 $\sum_{i=1}^{len}(len-i+1)val_i$ 即可。
- 时间复杂度 $O(n\log n)$，可以得到 $100Pts$ 的好成绩。