题解

whzzt

2018 年 6 月 9 日

安徽师范大学附属中学

差

在 $10^7 \times 10^7$ 的网格上有不超过 10^3 个关键点，67 次询问内确定到所有关键点距离和最近的点到所有关键点的距离和。

$q \leq 10^4$

直接提交样例程序，期望得分 2 分。

$q \leq 200$

实际上就是求 x 和 y 两维的中位数位置的答案。

$q \leq 200$

实际上就是求 x 和 y 两维的中位数位置的答案。

两维分别三分，询问次数 $4\log_{1.5} n$，期望得分 44 分。

$q \leq 100$

三分好蠢啊，直接二分斜率吧。

$q \leq 100$

三分好蠢啊，直接二分斜率吧。

询问次数 $4 \log_2 n$，期望得分 60 分。

$q \le 75$

两维分别二分好蠢啊，两维可以一起二分，这样三次询问就可以做一次二分了。

$q \le 75$

两维分别二分好蠢啊，两维可以一起二分，这样三次询问就可以做一次二分了。

询问次数 $3\log_2 n$，期望得分 $76 \sim 92$ 分。

$q \leq 70$

瓶颈在于每次都要做两次询问，回到三分的做法，可以令 $p = \frac{3-\sqrt{5}}{2}$，当区间为 $[x, y]$ 时我们将三分的两个端点分别设成 $x + p(y - x), y - p(y - x)$ ，可以发现这一次的某个端点的值下一次仍会用到。

$q \leq 70$

瓶颈在于每次都要做两次询问，回到三分的做法，可以令 $p = \frac{3-\sqrt{5}}{2}$，当区间为 $[x, y]$ 时我们将三分的两个端点分别设成 $x + p(y - x), y - p(y - x)$ ，可以发现这一次的某个端点的值下一次仍会用到。

不过这里询问是整数，我们可以将所有实数下取整。可以发现，31 次三分后区间内不会超过 5 个元素，同时我们已经有了区间内的 3 个元素 (除非在边界上，但这种情况下区间内的元素只有 4 个)，于是每一部分不会超过 34 次询问。

$q \leq 70$

瓶颈在于每次都要做两次询问，回到三分的做法，可以令 $p = \frac{3-\sqrt{5}}{2}$，当区间为 $[x, y]$ 时我们将三分的两个端点分别设成 $x + p(y-x), y - p(y-x)$，可以发现这一次的某个端点的值下一次仍会用到。

不过这里询问是整数，我们可以将所有实数下取整。可以发现，31 次三分后区间内不会超过 5 个元素，同时我们已经有了区间内的 3 个元素 (除非在边界上，但这种情况下区间内的元素只有 4 个)，于是每一部分不会超过 34 次询问。

只要合理取值就可以去掉一次多余的询问，于是总询问次数达到 67 次，可以通过。

阵

求 $n \times m$ 的 $0/1$ 网格中 0 连成一个联通块而所有 1 没有公共边的方案数。

题解

首先搜出所有合法状态，记录 0/1 的位和 0 的连通性，状态一共只有 2000 种左右，而转移只有 10^5 种左右。因此暴力跑出前 2000 个答案之后跑 Berlekamp-Massey 算法即可。

题解

首先搜出所有合法状态，记录 0/1 的位和 0 的连通性，状态一共只有 2000 种左右，而转移只有 10^5 种左右。因此暴力跑出前 2000 个答案之后跑 Berlekamp-Massey 算法即可。

根据不同的实现方式和不同的获得答案的方法可以得到不同的分数。

题解

首先搜出所有合法状态，记录 0/1 的位和 0 的连通性，状态一共只有 2000 种左右，而转移只有 10^5 种左右。因此暴力跑出前 2000 个答案之后跑 Berlekamp-Massey 算法即可。

根据不同的实现方式和不同的获得答案的方法可以得到不同的分数。

~~垃圾出题人卡我常数。~~

挑战

给定一张 n 个点 m 条边的弦图，求删去尽量少的点及其所连的边，使得图中不存在环。

弦图即任意一个长度不小于 3 的环都存在连接环上不相邻的节点间的弦的简单无向联通图。

挑战

给定一张 n 个点 m 条边的弦图，求删去尽量少的点及其所连的边，使得图中不存在环。

弦图即任意一个长度不小于 3 的环都存在连接环上不相邻的节点间的弦的简单无向联通图。

~~请一位挑战 npc 成功的同学上来讲一下题。~~

图中任意一个点至多属于一个环

任意顺序加点，能加就加即可。时间复杂度 $O(n)$，期望得分 $1 \sim 6$ 分。

图中任意一条边至多属于一个环

显然是一堆三角形套一个点上，把中间的点炸了就行了，期望得分 6 分。~~这组数据可能水到满足上一个 subtask 的条件~~。

$n \le 18$

暴力枚举或者随机排列尽量加，期望得分 $12 \sim 18$ 分。

$m \leq n + 20$

状压下下面所有还有返祖边的点的情况然后暴力 DP 即可，该算法期望得分 19 分。

对 $a \le b < c \le d$，若 $(a, d) \in E$，则 $(b, c) \in E$

我们可以将不存在环转化为每个团至多选择两个点。在这个子任务中，每个极大团是一个下标区间，我们可以找出所有的极大团，然后就转化为了一个区间内不能选择超过 2 个点。

对 $a \le b < c \le d$，若 $(a, d) \in E$，则 $(b, c) \in E$

我们可以将不存在环转化为每个团至多选择两个点。在这个子任务中，每个极大团是一个下标区间，我们可以找出所有的极大团，然后就转化为了一个区间内不能选择超过 2 个点。

我们从左至右考虑每个极大团，可以令 $f_{i,j,k}$ 表示第 i 个团中选择了 j 和 k 的最大答案。这样的总状态数是 $O(m\sqrt{m})$，因为一个极大团的状态数为 S^2，必然意味着当前比起上一个极大团多连了至少 S 条边。同时，状态数为 S^2 也意味着总边数为 S^2，因此 S 是 $O(\sqrt{m})$ 的，于是复杂度分配到每条边上，状态总数即为 $O(m\sqrt{m})$。

对 $a \le b < c \le d$，若 $(a, d) \in E$，则 $(b, c) \in E$

我们可以将不存在环转化为每个团至多选择两个点。在这个子任务中，每个极大团是一个下标区间，我们可以找出所有的极大团，然后就转化为了一个区间内不能选择超过 2 个点。

我们从左至右考虑每个极大团，可以令 $f_{i,j,k}$ 表示第 i 个团中选择了 j 和 k 的最大答案。这样的总状态数是 $O(m\sqrt{m})$，因为一个极大团的状态数为 S^2，必然意味着当前比起上一个极大团多连了至少 S 条边。同时，状态数为 S^2 也意味着总边数为 S^2，因此 S 是 $O(\sqrt{m})$ 的，于是复杂度分配到每条边上，状态总数即为 $O(m\sqrt{m})$。

而转移显然可以 $O(1)$，于是总的复杂度即为 $O(m\sqrt{m})$。

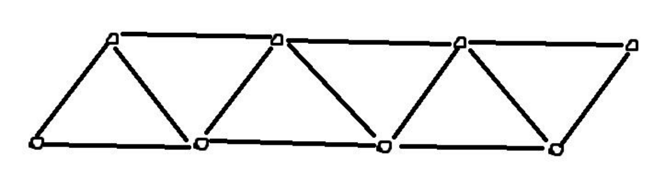
$n \leq 50$

~~模拟退火什么的有好好学吗？~~

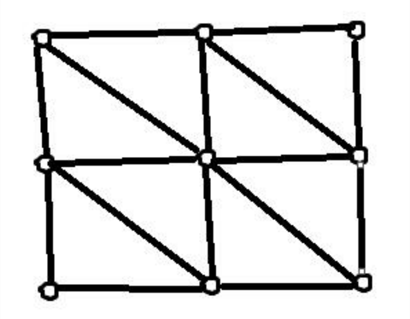
$n \leq 50$

~~模拟退火什么的有好好学吗?~~

emmm, 造了一组这样的图，能不能过就是你的造化了。



试说明下图是否是弦图：



$O(\text{poly}(n))$ 做法

我们考虑弦图的性质，一个弦图一定存在一个完美消除序列，而任意一个点与所有和它相邻且在完美消除序列中在它之后的点构成一个团：只要通过归纳我们就可以证明这一点。实际上，该定理的逆定理也成立。通过这样的方法我们能够找出所有的极大团。

$O(\text{poly}(n))$ 做法

我们考虑弦图的性质，一个弦图一定存在一个完美消除序列，而任意一个点与所有和它相邻且在完美消除序列中在它之后的点构成一个团：只要通过归纳我们就可以证明这一点。实际上，该定理的逆定理也成立。通过这样的方法我们能够找出所有的极大团。

求完美消除序列可以采用最大势算法 $O(n+m)$ 求得。

$O(\text{poly}(n))$ 做法

考虑之前解决若干团拼在一起的问题时，我们使用的是一个 $O(m\sqrt{m})$ 的 DP，可以考虑将其拓展到更一般的情况。

$O(\mathrm{poly}(n))$ 做法

考虑之前解决若干团拼在一起的问题时，我们使用的是一个 $O(m\sqrt{m})$ 的 DP，可以考虑将其拓展到更一般的情况。

我们可以猜想，所有极大团构成了一棵树的结构，满足任意两个团的交在它们路径上的所有团中都出现了，如果是这样的话，我们仍然可以套用之前的 DP。

$O(\mathrm{poly}(n))$ 做法

考虑之前解决若干团拼在一起的问题时，我们使用的是一个 $O(m\sqrt{m})$ 的 DP，可以考虑将其拓展到更一般的情况。

我们可以猜想，所有极大团构成了一棵树的结构，满足任意两个团的交在它们路径上的所有团中都出现了，如果是这样的话，我们仍然可以套用之前的 DP。

通过一些打表验证我们会发现这个结论是正确的，于是根据暴力实现的效率，可以通过 $n \le 50 \sim 5000$ 不等的部分分。

$O(\text{poly}(n))$ 做法

考虑之前解决若干团拼在一起的问题时，我们使用的是一个 $O(m\sqrt{m})$ 的 DP，可以考虑将其拓展到更一般的情况。

我们可以猜想，所有极大团构成了一棵树的结构，满足任意两个团的交在它们路径上的所有团中都出现了，如果是这样的话，我们仍然可以套用之前的 DP。

通过一些打表验证我们会发现这个结论是正确的，于是根据暴力实现的效率，可以通过 $n \leq 50 \sim 5000$ 不等的部分分。

如果实现得较为优秀可以同时通过前面所有的部分分，期望得分 $69 \sim 95$ 。

$m \leq 10^5$

之前建树的效率太低，我们需要一个构造性的方法。

$m \le 10^5$

之前建树的效率太低，我们需要一个构造性的方法。

然利用完美消除序列，我们将一个点包含在它所存在的最后一个极大团中，从后往前贪心。接下来考虑在碰到一个新的极大团时，我们将它接到其后面所有的节点所在的团中，深度最浅的一个团上。我们可以发现，对于所有不在当前团中的节点，这些节点一定在其父亲的团中。于是这样我们构造性地证明了之前结论的正确性，同时这个构造的时间复杂度为 $O(m)$。

之前建树的效率太低，我们需要一个构造性的方法。

然利用完美消除序列，我们将一个点包含在它所存在的最后一个极大团中，从后往前贪心。接下来考虑在碰到一个新的极大团时，我们将它接到其后面所有的节点所在的团中，深度最浅的一个团上。我们可以发现，对于所有不在当前团中的节点，这些节点一定在其父亲的团中。于是这样我们构造性地证明了之前结论的正确性，同时这个构造的时间复杂度为 $O(m)$。

建出树后我们只要直接树形 DP 记录方案即可，总的时间复杂度为 $O(m\sqrt{m})$。