分治相关题目选讲

jerome_wei

cdqz

HDU 6566

给 n 个点的树，需要选出一个独立集，使得 a_i 的总和是 m，且 b_i 最大，同时需要计算方案数。
$n \leq 100, m \leq 5000$。

HDU 6566

朴素 DP 是 $O(nm^2)$ 的。

HDU 6566

朴素 DP 是 $O(nm^2)$ 的。
考虑点分治，在点分树上面 dp，由于能和一个点相邻的点要么是点分数上的祖先，要么是子树中的点，因此可以在 DP 的时候记录点分树上的祖先被选择的情况。

HDU 6566

朴素 DP 是 $O(nm^2)$ 的。
考虑点分治，在点分树上面 dp，由于能和一个点相邻的点要么是点分数上的祖先，要么是子树中的点，因此可以在 DP 的时候记录点分树上的祖先被选择的情况。
这样就不需要合并，只有加入一个点的操作了。复杂度 $O(n * 2^{\log n} * m)$。

gym 102331F

大象有一个 n 个点的无向图，每个点有一个权值 w_i，一开始图是空的。
他有 n 个三元组 (a_i, b_i, c_i)，$1 \le a_i, b_i \le n, a_i \ne b_i, s_i \ge 0$。
之后他想要不断进行以下操作：

- 如果找不到 a_i, b_i 不在一个连通块里，并且 a_i 所在连通块的权值和 $+ b_i$ 所在连通块的权值和 $\ge s_i$，则结束。
- 否则找到最小的满足以上条件的 i 并且将这条边加入。
  $n \le 300000, 5s$

gym 102331F

由于每一条边是由两个连通块的大小去决定的，所以不好进行判断。
然而注意到如果只有一边的连通块会扩张，那么我们是很好对其进行维护的。

gym 102331F

由于每一条边是由两个连通块的大小去决定的，所以不好进行判断。
然而注意到如果只有一边的连通块会扩张，那么我们是很好对其进行维护的。
那么我们可以考虑将每条边拆分成两个单向的部分，每个部分的权值分别是 $c_i/2$，分别存储在两边。

gym 102331F

如果减少的时候某一个部分被删除成了 0，c_i 至少被减去了一半。所以我们在这个时候直接进行重构。维护出两边的结果即可。

gym 102331F

如果减少的时候某一个部分被删除成了 0，c_i 至少被减去了一半。所以我们在这个时候直接进行重构。维护出两边的结果即可。
我们用 set 维护当前连通块的所有边，全局加在结点上打一个标记，合并连通块的时候启发式即可。
总复杂度 $O(n \log n(\log n + \log v))$。

不知道哪里的题

给一个无向图，支持加边删边，求每个结点有多少个边双联通分量（保证图时刻联通）
$n, m \leq 200000$。

首先的想法是进行线段树分治，但是线段树分治的之后的实现是一个大问题。

首先的想法是进行线段树分治，但是线段树分治的之后的实现是一个大问题。
不难想到维护边双构成的树，我们可以考虑 dfs 到当前节点之后哪些位置是有用的。

首先的想法是进行线段树分治，但是线段树分治的之后的实现是一个大问题。
不难想到维护边双构成的树，我们可以考虑 dfs 到当前节点之后哪些位置是有用的。
不难发现只用考虑在子树中出现的节点构成的虚树，并且总点数是 $O(n \log n)$ 的。

那么我们可以在每次在虚数上加边，然后暴力跑一次缩点之后再进行一次缩点即可。
需要注意精细实现。
复杂度 $O(n \log n)$

CF 576E

给一张 n 个点 m 条边的无向图，初始时每条边没有颜色。q 次操作，每次操作将一条边染成 k 种颜色中的一种。如果一次操作后某种颜色的边构成的子图不是二分图，该操作不会被执行。判断每次操作是否被执行。

$n, m, q \leq 5 \times 10^5$

6s,600M

CF 576E

考虑离线怎么动态判定一张图加上一条边之后是否是二分图：线段树分治，插入的时候用并查集维护颜色。复杂度 $O(n \log^2 n)$
现在的问题是由于我们不知道每次是否执行，所以并不能直接离线。

CF 576E

考虑我们先默认每次操作都执行来确定每条边的起始与结束。
我们可以换一种方式考虑：我们看作每种操作都有执行（即我们确定了每条边的起始与结束）但是我们一开始并不知道其颜色。
考虑线段树分治的过程，每次分治的时候如果已经访问到了一个操作，这个这个操作的颜色实际上已经确定了下来，所以并不会影响线段树分治的操作。于是总复杂度 $O(n \log^2 n + nk)$。

JOISC2020 Day2 T1

交互题：有 $2N(N \leq 500)$ 只变色龙，N 只性别为 X 以及 N 只性别为 Y。
每个变色龙有一个原色：X 里面两两不同，Y 里面两两不同，并且每只变色龙能找到一个和他原色相同的。
每只变色龙都爱上了一只变色龙。关于恋爱对象的可公开情报如下：

- 每只变色龙都很专一于唯一一只异性的变色龙。
- 一只变色龙和它的恋爱对象的原色不同。
- 不存在两只变色龙同时追求另一只变色龙。

你可以询问一个集合的变色龙，对于一只在集合里面的变色龙 s，令 t 为它的恋爱对象。s 的肤色由以下方式决定：

- 如果 t 参加了这场会议，则 s 的肤色为 t 的原色。
- 如果 t 没参加这场会议，则 s 的肤色为 s 的原色。

然后你可以得到这个集合中的变色龙的肤色种类数

JOISC2020 Day2 T1

你最多能询问 20000 次，你想要知道所有相同原色的变色龙。
子任务：前 n 只性别为 X

JOISC2020 Day2 T1

首先考虑你已经知道性别的情况下：你可以考虑直接对于一个点进行分治查看是否存在颜色相同的情况：（是否递归下去的依据是颜色是否等于集合大小）

JOISC2020 Day2 T1

首先考虑你已经知道性别的情况下：你可以考虑直接对于一个点进行分治查看是否存在颜色相同的情况：（是否递归下去的依据是颜色是否等于集合大小）
这样对于每个 X 变色龙，你可以得到的是三个点：一个是入度，一个是出度，一个是颜色相等的。（或者只有颜色相等的）。

JOISC2020 Day2 T1

首先考虑你已经知道性别的情况下：你可以考虑直接对于一个点进行分治查看是否存在颜色相同的情况：（是否递归下去的依据是颜色是否等于集合大小）
这样对于每个 X 变色龙，你可以得到的是三个点：一个是入度，一个是出度，一个是颜色相等的。（或者只有颜色相等的）。
考虑已知这三个点的情况下：你考虑询问两个变色龙以及当前变色龙，发现只有：当前 变色龙 + 入度 + 同色 的答案是 1。于是可以找到出度。
于是二分图的情况就被解决了。

JOISC2020 Day2 T1

可以发现能够二分的条件实际上是单独查询除开这一个点的所有集合的时候颜色个数等于集合大小。
那么每次依次尝试能够加入则加入，如果不能加入的话，其实一定是与集合内部有边（或者同色）。
那么二分次数就是 $3N$ 的，其余的各种判断只需要花费 $O(N)$ 次操作，可以通过本题。

奇怪的题

给 n 个数 a_i，以及一个 n 次多项式 B，对于每个 i，求：

$$[x^n]\prod_{j!=i}(1-a_jx)B(x)$$

显然有一个简单的暴力做法：
考虑

$$[x^n]\frac{\prod(1-a_jx)B(x)}{1-a_ix}$$

容易发现答案是关于 a_i 的多项式，多点求值即可。

我们考虑将 B 系数翻转得到 $B'(x)$，记其系数为 b_i。
显然 i 位置的答案可以表示为 $\sum_i b_i[x^i](F_l * F_r)$
考虑分治 FFT，首先预处理区间的 F 的乘积。
考虑我们如何向下递归（向右为例）
$ans = \sum b[i] \sum F_l[j]F_r[i-j]$
将 b 和左边的多项式减法卷积一下，只需要保留 $n/2$ 项即可。

luogu P5655

给定长度为 n 的数组 a，Q 次询问一个区间的 lcm。
由于输出较大，你只需要输出答案对 10^9+7 取模的值。
$n \leq 300, Q \leq 300, T \leq 300$

luogu P5655

最暴力的做法显然是直接对 a_i 算与之前的数的 lcm 的答案，将之前的 lcm 写成之前每次增大数 b_i 的乘积的形式，可以先对 a_i 取模之后求 gcd。

luogu P5655

最暴力的做法显然是直接对 a_i 算与之前的数的 lcm 的答案，将之前的 lcm 写成之前每次增大数 b_i 的乘积的形式，可以先对 a_i 取模之后求 gcd。

考虑分治，先将前后缀求出来记为 b，我们现在需要算的是 $\gcd(\prod_{ql}^{mid} b_i, \prod_{mid}^{qr}, b_i)$ 。

luogu P5655

显然，我们暴力枚举两个数的 gcd ，可以做到 $Tn^2 * \log V$。

luogu P5655

显然，我们暴力枚举两个数的 gcd ，可以做到 $Tn^2 * \log V$。
能不能做到更优?
考虑那个 $\log V$ 其实因为 gcd 每次都跑满了。我们想要让它变成均摊下来总共一个 log。

luogu P5655

我们记录 c_i 为 $[\mathrm{mid}+1, i]$ 的 b 的乘积。ql 从 $\mathrm{mid}-1$ 移动到 l 的时候，相当于每次在所有右边部分之前加入一个 b_l 。我们需要求出新的 b'_i 。
考虑 $\mathrm{lcm}(b_l, c_{i+1})/\mathrm{lcm}(b_l, c_i) = b'_i$ ，简单化简，有：

$$\frac{b_{i+1}}{\gcd(c_{i+1}, b_l)/\gcd(c_i, b_l)} = b'_i$$

考虑 $\gcd(c_{i+1}, b_l) = \gcd(c_{i+1}, c_i, b_l)$。
我们可以用 $\gcd(c_i+1, b_l)$ 和 c_i 取最大公因数，即可算出新的 $\gcd(c_i+1, b_l)$。
可以分析出这里是每次移动 l 均摊一个 log 的。
于是复杂度变成了 Tn^2 。

找不到例题就随便胡了个题面

我们有 n 个物品，每个物品有一个大小和权值，现在查询用区间的物品做大小为 W 的背包能组成的背包的一个位置的值。查询的区间互相不包含（可以相交）
$n, m \leq 10^5, W \leq 100$。

最简单的想法是直接分治，这样会带一个 log。

最简单的想法是直接分治，这样会带一个 log。
然而实际上我们可以避免分治：
我们考虑现在实际上操作只有向后面加一个物品，或者在最前面删除一个物品。

最简单的想法是直接分治，这样会带一个 log。
然而实际上我们可以避免分治：
我们考虑现在实际上操作只有向后面加一个物品，或者在最前面删除一个物品。
这样的话我们仍然考虑分成两半，在加的时候暴力加，左边的部分我们可以维护每个后缀的答案。

最简单的想法是直接分治，这样会带一个 log。
然而实际上我们可以避免分治：
我们考虑现在实际上操作只有向后面加一个物品，或者在最前面删除一个物品。
这样的话我们仍然考虑分成两半，在加的时候暴力加，左边的部分我们可以维护每个后缀的答案。
我们在删除的时候就可以直接不管，如果左边全部删完了，就把右边的部分重构成左边的。
这样每个位置只会被重构一次。
所以总复杂度 $O((n+m)W)$。