

Azshara

题目说被点名的玩家需要跑到最外侧的木桩上，因此我们只需要对最外侧的木桩进行连边，找到最外侧的点就是求个凸包，要先对所有点跑一个凸包，只对凸包上的点进行操作。

之后我们对于凸包上的所有点编号。假设凸包上有 m 个点，我们按照逆时针或者顺时针的顺序编号为 0 到 $m-1$ 。

由于点数只有 400 ，并且凸包上的点数是很大概率不到 400 ，我们可以枚举凸包上的所有不相邻的两点 i, j 之间的连线，对于每一条连线我们可以枚举每一个圆是否与它相交或者相切，并将答案记录一个二维数组中 $mp[i][j]$ 中。注意这里要判断凸包上的第一个点和最后一个点，因为他们在图形上也是相邻的。还有一个就是在判断的时候需要把精度调整到 $1e-10$ 来判断。

之后由于线段之间不可相交并且要求出最大的线段数，我们需要区间dp。因为我们对于凸包上的点按照顺序排序，可以根据点的编号来判断是否相交。直接枚举区间起点和终点，再枚举中间断点就可以了。

Blow up the city

首先建立反向图，将城市到若干个终点看成从若干个起点出发到某个城市，再用一个源点连接那些度为 0 的点，即可看成从源点出发到某个城市。要炸掉一个点使得无法到达某个城市，那么需要炸掉的是从源点到该城市的必经点，考虑建立支配树，根据定义可知支配树到根的链上结点个数就是必经点的个数。两个城市的容斥减去LCA到根上这条链即可。由于保证是DAG，因此直接按拓扑序建树即可，建完树利用结点的depth来求点到根的链长，注意最后答案要减去一开始添加的源点。

Yukikaze and Demons

题意：给出一棵树，每个顶点有一个十进制数位，定义一条简单路径 (u, v) 的权值为将从 u 到 v 的路径上的所有顶点上的数位连起来所组成的十进制数。统计有多少条路径的权值能被给定的数 k 整除。

分治统计答案。

设当前分治块中心为 r ，需要统计经过 r 的满足条件的路径数量。

将每条经过 r 的路径 (u, v) 分成两段，第一段为 (u, r) ，第二段为 (r, v) 去掉 r ，做法为枚举第二段并统计拼上第二段后满足条件的第一段路径有多少条。

不妨设当前枚举的第二段权值为 y ，且 $dis(r, v) = h$ ，即第二段的权值是一个有 h 位的十进制数 y ，则第一段的权值 x 必满足同余方程 $10^h x + y \equiv 0 \pmod{k}$

解这个同余方程等价于解二元一次不定方程 $10^h x + mk = k - y$ ，可利用扩展欧几里得解决，无解则可直接跳过该节点。

设解出 $x \equiv x_0 \pmod{k'}$ ，暴力统计大概率会 T ，所以需要预处理出可能被用到的答案。

观察到 $k' = \frac{k}{\gcd(k, 10^h)}$ ，即至多只有 $\lfloor \log_2 k \rfloor$ 种不同的 k' ，因此对每种 k' 预处理出对应的答案即可。

最终的时间复杂度为 $O(N \log N \log k)$ ，可能有点卡常。

Distribution of books

题目要使得最大值最小？考虑二分答案

对于每次二分答案（假设为 x ），如何判定 x 能否满足分为 k 份的要求呢？考虑动态规划

令 $dp[i]$ 表示前 i 个数最多能分成几段，则

```
dp[i] = max(dp[j]) + 1;
(sum[i] - sum[j] <= x)
```

如果直接dp, 时间复杂度为 n^2 , 显然会TLE!

考虑用平衡树维护或者离散化后权值线段树维护, 总体复杂度 $n \cdot \log(n)$

Easy Math Problem

我们设答案 $res = \sum_{i=1}^n \sum_{j=1}^n gcd(i, j)^k lcm(i, j) [gcd(i, j) \in prime]$, 那么我们先枚举 d 作为 gcd , 式子变为

$res = \sum_{d=1}^n [d \in prime] \sum_{i=1}^n \sum_{j=1}^n d^k lcm(i, j) [gcd(i, j) = d]$, 考虑到 $i * j = gcd(i, j) * lcm(i, j)$, 继续变换得: $res = \sum_{d=1}^n d^k [d \in prime] \sum_{i=1}^n \sum_{j=1}^n \frac{i * j}{d} [gcd(i, j) = d]$, 因为 $gcd(i, j) = d$, 因此 i, j 都是 d 的倍数, 那么我们可以得到 $res = \sum_{d=1}^n d^{k+1} [d \in prime] \sum_{i=1}^{\frac{n}{d}} \sum_{j=1}^{\frac{n}{d}} [gcd(i, j) = 1]$.

我们对后面的式子范围除 d 可得 $res = \sum_{d=1}^n d^{k+1} [d \in prime] \sum_{i=1}^{\frac{n}{d}} \sum_{j=1}^{\frac{n}{d}} ij [gcd(i, j) = 1]$.

对于 $\sum_{i=1}^n \sum_{j=1}^n ij [gcd(i, j) = 1]$, 求值范围等价于将范围拆成 i, j 各自为其上限以及相等时的范围:

$[1 \leq i \leq n][1 \leq j \leq n] = [1 \leq i \leq n][1 \leq j < i] + [1 \leq j \leq n][1 \leq i < j] + [i = j][1 \leq i \leq n]$. 这个第一部分相当于枚举1到 n 的 i , 然后枚举小于 i 的 j , 因为 $gcd(i, j) = 1$ 的限制, 因此我们其实是在对每个值求小于这个值且与其互质的数的和, 这是一个经典的求和公式, 我们有

$\sum_{j=1}^{i-1} j [gcd(i, j) = 1] = \frac{i\varphi(i) + [i=1]}{2}$ 因此整个第一部分化简后为 $\sum_{i=1}^n \frac{i^2\varphi(i) + i[i=1]}{2}$, 因为第二部分除了求和符合和第一部分相同, 因此一二部分和为 $\sum_{i=1}^n i^2\varphi(i) + i[i=1]$, 对于第三部分, 当 $i > 1$ 时 $gcd(i, j) = i > 1$, 因此没有大于1的贡献, 我们将第三部分加上刚好消掉一二部分 $i[i=1]$ 这个限制, 因此整个后面的式子结果为 $\sum_{i=1}^n i^2\varphi(i)$.

经过上面的分析, 最后答案形式为 $res = \sum_{d=1}^n d^{k+1} [d \in prime] \sum_{i=1}^{\frac{n}{d}} i^2\varphi(i)$. 那么对于式子我们可以分为两个部分分析, 对于第一部分我们设 $g(n) = \sum_{d=1}^n d^{k+1} [d \in prime]$, 这个式子就是在求1到 n 里面质数的 k 次幂和, 我们直接用Min25筛中求质数部分和的过程就能完成统计, 而求 $k+1$ 次方和可以用拉格朗日插值法处理, 每次求和复杂度是 $O(k)$, 而我们整除过程中有 $O(\sqrt{n})$ 个值要求和, 因此这个过程的复杂度是 $O(k\sqrt{n})$. 现在考虑第二部分, 我们设 $S(n) = \sum_{i=1}^n i^2\varphi(i)$, 我们可以直接暴力Min25求, 但是这个式子用杜教筛处理更加优美.

按照杜教筛的套路得到 $t(i)S(n) = \sum_{i=1}^n (f * t)(i) - \sum_{i=2}^n t(i)S(\frac{n}{i})$, 因为 $(f * t)(i) = \sum_{d|i} f(d)t(\frac{i}{d})$, $f(d) = d^2\varphi(d)$, 那么我们令 $t(i) = i^2$, 因此 $(f * t)(i) = \sum_{d|i} \frac{d^2\varphi(d)i^2}{d^2} = \sum_{d|i} i^2\varphi(d) = i^3$, 而 $t(1) = 1$, 因此最终 $S(n) = \sum_{i=1}^n i^3 - \sum_{i=2}^n i^2 S(\frac{n}{i})$.

那么我们综合两个部分就能得到答案了, 因为Min25筛复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$, 而我们预处理出 $n^{\frac{2}{3}}$ 范围的前缀和后, 杜教筛复杂度为 $O(n^{\frac{2}{3}})$, 因此整体复杂度是 $O(\frac{n^{\frac{3}{4}}}{\log n} + n^{\frac{2}{3}} + k\sqrt{n})$.

Fansblog

威尔逊定理 + 质数的密度分布。

upd: 万分抱歉, 从敝校OJ搬题目到HDOJ时, 按要求将所有题目数据改为单文件, 然而题面没有及时更新导致出锅。

Find the answer

对于第 i 个位置, 怎样选择数字才会使满足条件下选择数字数目最少呢? 很容易想到, 需要选择前 $i-1$ 个数中较大的数字, 使其变为0

基于这个思想, 如果我们对于每个位置 i 都暴力去找最大的前几个数, 显然会TLE!

可以注意到, 题目可以转化为前 $i-1$ 个数中最多选出多少个数字和 $W[i]$ 相加使得其和小于等于 m (很容易想到, 选择较小的数才会使选的数最多)。转化之后就很容易想到用线段树来维护了。

我们对给定数组进行离散化, 对于离散化之后的数组建立一颗线段树, 线段树上的每个节点记录区间之和以及区间内数字个数。时间复杂度: $N \cdot \log(N)$

Game

先对序列求前缀异或和。修改操作相当于对前缀异或和的单点修改, 查询相当于询问区间相同点对数, 故可以用三维莫队解决问题。

K subsequence

题意转换一下就是每次给定一个序列, 我们可以拿 k 个单调不减子序列, 每次拿了之后我们可以获得这个单调不减子序列的权值和并且在之后不能再继续拿走这些数了. 问拿了 k 次之后能够得到的最大权值和.

考虑这个问题和经典网络流问题“最长不下降子序列问题”相似, 我们可以考虑对这个建图并且运用网络流的做法解决这个问题. 因为每次我们要的是权值和, 因此这个题的做法就可以确定是费用流做法. 下面考虑如何建图.

首先我们要保证只能拿 k 次, 那么我们直接从源点出发, 将源点拆为入点 s_0 和出点 s_1 , 并且限制 s_0 到 s_1 的流量为 k , 费用为0, 这样我们就保证了只拿 k 次的条件。而对每个点, 我们采用同样的方式, 对于 a_i 这个点, 将其拆为 a_{i0} 和 a_{i1} 两个点, 流量为1表示只能选一次这个点, 费用为 $-v_i$ 表示能够得到的费用. 然后对于每个 i , 我们枚举 $j > i$, 如果 $v_j \geq v_i$, 那么我们就从 a_{i1} 到 a_{j0} 连一条流量为1并且费用为0的边.

最后我们将 s_1 与每个点的入点 a_{i0} 连一条流量为1并且费用为0的边, 然后从每个点的出点 a_{i1} 向汇点 t 连一条流量为1并且费用为0的边, 然后我们从 s 到 t 跑一次最小费用最大流并且取反就是我们的答案。

update: 由于出题人用的是 $Dijkstra$ 优化的费用流模板, 没想到卡掉了很多人的 $SPFA$ 费用流模板, 实在是非常抱歉。

Sindar's Art Exhibition

我们先考虑序列上的这个问题, 并且假设所有的 y 值相同, 那么假设询问从 l 到 r , 初始有 x 幅画, 即是要求一个以 x 为首项, $-y$ 为公差的等差数列乘上对应 $f[i]$ 的和, 如果线段树下标为 i 的位置像通常一样维护 $f[i]$ 是很难解决的。但可以发现对于 l, r 相同的两个询问, 等差数列的每一项差值都相同, 如果差值为 k , 那么和就差了 $k * \sum_{i=l}^r f[i]$, 只要知道其中一个询问的答案, 通过维护 $f[i]$ 的前缀和就能快速算出另一个询问的答案, 但这个做法要求对每个区间都算出一个询问的答案, 显然效率低下。考虑通过构造, 使得线段树上询问 $[l, r]$ 区间即可得到一个该区间的初始答案。不难发现, 维护 $g[i] = i * y * f[i]$ 的数组可以实现。再略加思索, 发现 y 并不是要一定相等的, 只要保证每个位置的 y 不随询问变化即可, 维护的数组也就变成 $g[i] = i * sumy(i) * f[i]$ ($sumy$ 表示 y 数组前缀和), 序列上的问题就解决了。(当然 $f[i]$ 的和也是需要维护的。) 和其他问题类似, 可以通过树链剖分将树上问题转化为序列问题, 只是因为从 u 到 v 与从 v 到 u 的询问并不等价, 要考虑的情况更复杂一些, 但基本思想是一样的。

Squirrel

此题做法很多。可以先考虑不删边版本, 题意化简为求每个点到最远叶子距离, 一次dfs找到每个点向其子树方向走到叶子的最远距离 $f[i]$ 和次远距离 $s[i]$, 第二次dfs更新向父亲方向走的最远路径 $fa[i]$, 对于每个点的答案就是 $\max(f[i], fa[i])$ 。在考虑允许一次删边, 就在DP多开一维表删边与否, 并记录子树方向到叶子的最大, 次大, 第三大距离和向父亲方向走的最远距离。需要注意的是更新 $fa[i]$ 的时候要判断该节点是否是其父亲节点往叶子方向走最远, 次远路径上的点, 分情况转移即可。

