中国计算机学会 China Computer Federation

有根树（tree）

命题人：MIT 毛啸

讲题人：THU 高闻远

题意

- 给定一棵有根树 $T=(V,E)$，要求维护一个点集 S，操作有插入或删除单点。对集合 S中的点 u 定义 w_u 为 S 与其子树的交集大小。对每次操作后求一个最优的包含根的连通块 C，使得 $\max(a,b)$ 最小，a 是 C 与 S 的交集大小，b 是 $V-C$ 与 S 的交集中 w 最大值。
- $|V| \le 5\times10^5, q \le 10^6$。

题意转化与部分分

- 题目可以转化成选择 S 的一个子集，并且它们与根可以形成一个连通块。
- 进一步观察 a,b，可以发现我们选择的点一定是 w 最大的那些点。
- 于是我们就有了暴力做法：求出 w 的所有值，将它们排序求解，根据实现方式得到 0~30 分。
- 链的点答案为 |S|/2。

- 我们考虑优化下面这个暴力做法。

中国计算机学会
China Computer Federation

暴力做法

- 令我们选的 S 的子集为 A，$B = S - A$，将它们中元素的 w 分别装入两个桶中 $bucketA, bucketB$。
- lim 表示我们当前考虑的 w 值，初始时可以认为它是极大值，它最后代表集合 B 中的最大值。
- 要么$lim - 1$，要么$|A| + 1$，最后两个值将相等，即所求答案。

```
B = S
While (|A| < lim)
    If bucketB[lim] is empty:
        --lim
    Else:
        top = bucketB[lim].top()
        将top从B和bucketB中移出，加入A和bucketA
        ++|A|
```

暴力做法

```
|A| < lim:
    If bucketB[lim] is empty:
        --lim
    Else:
        top = bucketB[lim].top()
        将top从B和bucketB中移出，
        加入A和bucketA
        ++|A|
```

```
|A| > lim:
    If bucketB[lim] is empty:
        ++lim
    Else:
        top = bucketA[lim].top()
        将top从A和bucketA中移出，
        加入B和bucketB
        --|A|
```

- 从上面做法可以看出，答案每次变化至多为1。
- bucket不好维护，但我们可以不维护完整的bucket，只要保证top的值正确。
- 考虑只维护可能成为top的候选值，其他值不存入桶中。

做法

- 若一个点在 A 中，且它子树中有点也在 A 中，则它不可能是 $bucketA$ 的候选。同理，一个点在 B 中，且它有祖先也在 B 中，则它不可能是 $bucketB$ 的候选。
- 于是每个点到根路径上 $bucketA, bucketB$ 候选都至多一个。它们可以看成路径的分界点。
- 分界点上移很好处理，但如果下移则很麻烦，所以还是不能暴力处理。
- 考虑轻重链剖分，维护每条重链的候选，加入或删除时只有至多两条链会有分界点的变动，其他链只有 w 值的 +-1 变动。
- 对于未来才会使用到的候选，我们只需要维护它的 w 值，当 lim 变动时再调整分界点。

做法

- 将重链上 S 中的结点按深度排序，那么每次分界点移动都是前驱或后继的移动，于是我们可以使用 set 来维护。
- $bucket$ 用链表维护可以做到 w 变动时 $O(1)$ 维护好。
- $bucket$ 和重链候选点 w 值的更新可以单次 $O(1)$ 完成，总操作数 $O(q\log n)$。
- 新点与新分界点的 w 值可以树状数组完成，单次 $O(\log n)$，总操作数 $O(q)$。
- 做法流程即更新 w 值 -> 更新分界点与候选桶 -> 调用前面的子程序。
- 总时间复杂度 $O(q\log n)$。

中国计算机学会
China Computer Federation

猜数游戏

中国人民大学 王成瑞

题目大意

- 对于一个给定的剩余类集合 T，定义 $f(T)$ 为小 J 所选的数集为 T 时的最小游戏轮数。
- 已知一个模数 p 和模 p 意义下一些剩余类组成的集合 S。
- 求 $f(T)$ 的期望值，其中 T 为 S 的任一非空子集。

关于题意的疑问

- “期望的最小值”和“最小值的期望”是两个语义不同的偏正短语。
- 对于误解的题意，其描述应为“在采取最优策略的情况下询问次数的期望”。
  - 这是一个固定的值，并不存在“最小”的概念。
  - 事实上，很多OI题的题面也是如此描述的。

初步分析

- 定义：集合 S 是集合 T 的生成集，当且仅当 S 满足：
  - $S \subset T$
  - $\forall a \in T$，$\exists x \in S, k \in \mathbb{N}^+$ 使得 $a \equiv x^k (mod\ p)$

- 根据游戏规则，不难发现对于给定的 T，$f(T)$ 的值等于其最小生成集的大小。

算法一

- 暴力枚举 S 的所有非空子集 T。
- 暴力枚举 T 的所有子集检查其是否可以生成 T。

- 时间复杂度：$O(3^n n^2)$
- 可以通过测试点 $1 \sim 4$，共计 20 分。

算法二

- 对于特殊性质B

- 注意到数列 u^i 在模 p 意义下很快会归零，且相互表示关系构成了若干条链。
- 链之间对答案贡献独立，且仅与链的长度有关。
- 预处理答案即可。

- 可以通过测试点 10,16，共计 10 分
  - 结合前述算法可得 30 分

中国计算机学会
China Computer Federation

考虑建图

- 对于 $u,v \in S$，从 u 向 v 连边当且仅当存在正整数 k 使得
$$u^k \equiv v \pmod p$$
- 考察 $f(T)$ 在所建的图上的意义
  - 最小的“可到达” T 中所有点的集合
- 对于给定的 T，点 u 在其最优询问方案中当且仅当
  - $u \in T$
  - 可到达 u 的所有点均不在 T 中
- 因此可以分别计算图中每个点对答案的贡献。这部分的复杂度是 $O(n^2)$。

算法三

- 续前页，考虑如何确定图中的边。
- 对于 $u \in S$，暴力枚举 u^k
  - $(u,p)=1$：枚举次数不超过 $\varphi(p)$
  - $(u,p)>1$：枚举次数为 $O(\log p)$
- 时间复杂度：$O(np+n^2)$
- 可以通过测试点 $1 \sim 6$，共计 30 分。
  - 常数优秀者可能能通过测试点 $1 \sim 10$。

算法四

- 算法二中对非互素部分的处理已经达到了可以接受的复杂度，下面只考虑优化互素部分的建图。
- 仍然是枚举 $u \in S$，用 BSGS 算法检查 u 的所有出边
- 时间复杂度：$O(n^2\sqrt{p})$
- 可以通过测试点 $1 \sim 10$，共计 50 分。

原根

- 算法三最大的问题是 BSGS 算法求离散对数时每换一次底就要重新预处理一次，而“原根”则可以解决这一问题。
  - 特殊性质A

- 对于正整数 p，若存在 g 满足

$$g^i \quad (1 \le i \le \varphi(p))$$

在模 p 意义下两两不同余，则称 g 是 p 的一个原根。

- 模 p 意义下原根存在的充要条件？

中国计算机学会
China Computer Federation

算法五

- 考虑两个数 u, v，其在某一原根下的指标分别为 a, b

$$v \equiv u^k (mod\ p) \quad \Leftrightarrow \quad ak \equiv b \left(mod\ \varphi(p)\right)$$

- 由此可知 u 向 v 连边的充要条件是 $\gcd\left(a, \varphi(p)\right)$ 整除 $\gcd\left(b, \varphi(p)\right)$。

- 找到一个原根，并求出 S 中所有数的指标（离散对数），记作 $b_1, \cdots, b_n$。
- 按照 $\left(b_i, \varphi(p)\right)$ 之间的整除关系建图。

- 时间复杂度：$O\left(n\sqrt{p} + n^2\right)$
- 可以通过测试点 $1 \sim 20$，共计 100 分。

中国计算机学会
China Computer Federation

课后习题

1. $n \leq 10^5$，$p \leq 10^{14}$
2. 取消对模数 p 的所有限制，并将其范围扩大至 10^{18}

小结

- 本题综合考查了选手对数论、简单的图模型以及简单的计数原理的掌握
- 难度不是很大的签到题，但在具体讨论时仍有少量细节需要注意
- 希望本题能为各位选手理想成绩的取得助一臂之力

谢谢大家

选 课

中国人民大学 陈志朋

题目描述

- 一共有m种分类，第i种分类有n_i个课程，需要修够s_i个学分（可以超过）。学习每个课程能够得到w_i的学分，需要消耗c_i的脑力。总共要修满T个学分（可以超过），求最小消耗的脑力值。
- 附加关系（每条关系涉及的课程不一定在同一种分类中，各种关系在输入数据中给出）：
  - 两门课同时修，可能减少脑力消耗，可能增加消耗，也可能不能同时修。
- 无解输出-1。

$N \leq 15$

- 暴力枚举每门课程选或不选。
- 对于每一种选课状态，枚举每一条限制判断是否成立。
- 如果当前选课状态符合每一条限制，那么更新答案。
- 时间复杂度：$O(N^2 \times 2^N)$。

$N \leq 1000, p = 0$

- 对每一个类别的课程分开处理。
- 设$f[i][j]$表示第i个分类中，选择了j学分的课程，所需要花费的最少的脑力值。在每一个分类中，对f数组的处理是一个01背包的过程。转移方程如下：

$$f[i][j] = \min\{f[i]\left[j - w_{i,k}\right] + c_{i,k} | 1 \leq k \leq n_i\}$$

- 设$g[i][j]$表示前i个分类中，一共选择了j学分的课程，所需要花费的最少脑力值。转移方程如下：

$$g[i][j] = \min\{g[i-1][j-k] + f[i][k] | s_i \leq k \leq T + 2\}$$

- 时间复杂度：$O(N^2)$。

$w_i = 1, p = 0$

- 由于没有特殊关系，同时每门课程的学分都是1学分，可以采用贪心的选择方法。
- 对于第 i 个分类，选择花费脑力值最小的 s_i 门课程。同时，把剩余的课程放入一个小根堆中。
- 最后，从小根堆中选取最小的 $T - \sum_1^m s_i$ 门课程。
- 时间复杂度：$O(NlogN)$。

剩下的两个部分分

- 对于一下的两个部分分：
- $\mathrm{N} \le 10^4, C \le 50$，且有关系的课程在同一分类中。
- $\mathrm{N} \le 5\times10^4, C \le 50$。
- 基本解法与正解相同，但缺少优化，讲解正解时一起讲解。

正解

- 把每种分类分开处理，把有涉及特殊关系的课程的分类与其他分类分开处理。
- 先处理不涉及特殊关系的课程和分类（如果一个分类中有特殊关系的课程，那么只需把该课程独立出来，不在当前阶段处理即可）。
- 在一个分类中，可以对每一种学分的课程按消耗的脑力值从小到大排序。那么，一种选课的状态可以被抽象为3个数字，分别表示每种学分课程的选了几门。

正解

- 设$f[i][j]$表示第i个分类中，至少学习j学分的课程，所需要消耗的最小脑力值。
- 注意到，在每个分类中，只有$f[i][s_i-\alpha \ldots s_i+T-\sum_1^m s_i+2]$有用（$\alpha$表示该分类中，有特殊关系的课程的总学分）。于是，我们针对每一个$f[i][j]$进行贪心处理（需要通过离散化来开数组。为了方便表述，这里仍使用**未离散化**的方式）。
- 对于一个特定的学分j，我们先快速构造一组合法的选课方案：能选3学分的就选3学分的，然后能选2学分的就尽量选，然后全部选1学分的。如果还剩余1学分没有分配（显然不会有超过1学分未分配），那么少选一门3学分的课程，并且多选两门2学分的课程即可处理这个问题。

正解

- 得到一组合法的解之后，我们通过贪心调整的方法来寻找最少的脑力值消耗。
- $3 \to 1+1+1; 2 \to 1+1; 1+1 \to 2; 1+1+1 \to 3; 3+3 \to 2+2+2; 2+2+2 \to 3+3;$
- $1+2 \to 3; 3 \to 1+2; 1+3 \to 2+2; 2+2 \to 1+3$。
- 共10种调整方法，可以证明这样能够找到最优解。
- 只需证明一个状态通过变劣再变优的方案，都能通过上述调整方法，找到一种一直变优的方案，即可证明从任意一种状态开始，用上述的调整方法能够找到最优解。
- 对于两步的情况，枚举每两种方案的情况，分类讨论证明即可。
- 由于两步的情况能够得到转换，所以可以推出多步的情况也能得到转换。

正解

- 引理：如果一种调整方法会导致结果更劣，那么进行若干次，结果只会越来越劣。
- 引理证明：减少的课程的脑力值消耗会减少，增加的课程的脑力值消耗会增加。由单调性可知，总消耗值会越来越劣。
- 我们对调整方案进行分类：1、一种课换成另一种课；2、一种课换成两种课；3、两种课换成一种课。

$ba \to ab$; $cb \to bc$

$\Rightarrow a-b,\ b+a-c,\ c+b$ 更优

i) b是1学分的课，~~那么相当于~~

a是3学分，c是2学分 $\Rightarrow 3 \to 1+2$

$a=2$, $c=3$ $\Rightarrow 1+2 \to 3$

正解

ii) ⓑ $= 2$

ⓐ $= 1$, ⓒ $= 3 \Longrightarrow 1+1+2+2 \longrightarrow 3+3$.

t_2 / t_1 — 3学分　　P_3 / P_2 / P_1 — 2学分　　q_2 / q_1 — 1学分

~~假设 $P_3 \to q_1 + q_2$ 且~~ $t_1 + t_2 < P_1 + P_2 + q_1 + q_2$ (已知).

若 $P_2 + q_2 < t_1$, 则 $P_1 + q_1 < t_2$.

违背已知, 所以可以拆分为 $1+2 \longrightarrow 3$

$1+2 \longrightarrow 3$.

~~iii) ⓑ $= 3$~~

ⓐ $= 3$, ⓒ $= 1 \Longrightarrow 3+3 \longrightarrow 1+1+2+2$.

③ t_2 / t_1　② P_2 / P_1　① q_2 / q_1

若 $t_2 < P_1 + q_1$, 则 $t_1 < P_2 + q_2$.

~~违背~~矛盾. ∴ 可拆分为 $3 \longrightarrow 1+2$, $3 \longrightarrow 1+2$.

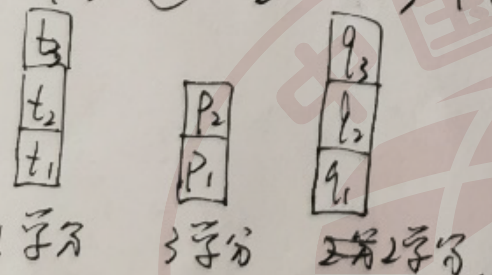
iii) ⓑ $= 3$

正解

∴ 拆分为 $3 \to 1+2$，$3 \to 1+2$.

iii) ⓑ $=3$

ⓐ $=1$，ⓒ $=2$ $\Rightarrow 1+1+1+3 \to 2+2+2$



① 若 $t_3+p_1 < q_1+q_2$

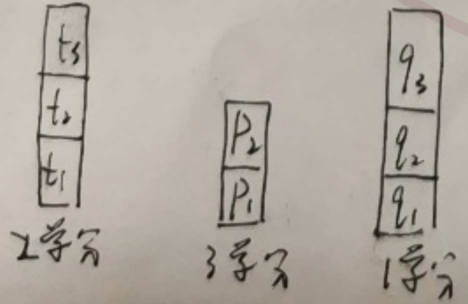
要满足条件，则必须 $t_1+t_2 > q_3$

那么 $t_2+t_3 > t_1+t_2 > q_3 > q_1$

∴ 拆分成 $1+1 \to 2$，$1+3 \to 2+2$.

否则，拆分成 $1+3 \to 2+2$，$1+1 \to 2$.

ⓐ $=2$，ⓒ $=1$ $\Rightarrow 2+2+2 \to 3+1+1+1$



由②知得 $t_1+t_2+t_3 > p_1+q_1+q_2+q_3$.

且 $t_1+t_2+t_3 < p_1+p_2$.

若 $t_2+t_3 < p_1+q_1$，要满足条件，则必须 $t_1 > q_2+q_3$.

∴ $t_3 > q_1+q_2$，∴ 拆分成 $2 \to 1+1$　$2+2 \to 3+1$

正解

- 生成$f[i][j]$的方案时，可以直接从$f[i][j+1], f[i][j+2], f[i][j+3]$快速得到。
- 从$f[i][j+1]$继承：$3 \to 2; 2 \to 1; 3 \to 1+1; 2+2 \to 3$; 直接减少一门已选的1学分的课程。
- 从$f[i][j+2]$继承：$3 \to 1; 3+3 \to 2+2$; 直接减少一门已选的2学分的课程。
- 从$f[i][j+3]$继承：直接减少一门已选的3学分的课程。
- 可证，若$f[i][j]$继承后未进行调整的结果比被继承的状态更劣，那么$f[i][j]$一定比被继承状态更劣。所以$f[i][j]$一定和被继承状态的值相同。
- 证明过程，对每一种继承方法分类，结合之前的结论，对所有调整方法进行讨论即可。

正解

小强行通过 S[3]-1，S[2]+1 来构一种情况。

考虑所有调整策略。

① 1+1 → 2 X (S[1]=0)

② 2 → 1+1 X (S[3]-1, S[1]+2)

③ 1+1+1 → 3 X (S[1]=0)

④ 3 → 1+1+1 X (S[3]-1, S[1]+2 更优)

⑤ ~~2+2 → 3+3+3~~ 2+2+2 → 3+3 X (S[2]-1, S[3]+1).

⑥ 3+3 → 2+2+2 (此处3的消耗小于2)

⑦ 1+3 → 2+2 X (S[1]=0)

⑧ 1+2 → 3 X (S[1]=0)

⑨ 3 → 1+2 X (此处3的消耗小于2)

⑩ 2+2 → 1+3 X (相当于 S[2]-1, S[1]+1).

正解

- 由于分类的顺序并不重要，我们对分类重新排序，使 $1\dots q$是均 未涉及特殊关系课程的分类，$q+1\dots m$是涉及特殊关系课程的分类。
- 设$g[i][j]$，表示前i个分类，至少比当前要求多学了j个学分所需要消耗的脑力值。当前要求指至少学满$\sum_1^i s_k$学分。
- $g[i][j]=\min\{g[i-1][j-k]+f[i][k+s_i] \mid 0\le k\le T-\sum_1^m s_i\}$
- 设$h[z]$，表示特殊关系的课程学了z状态（状压），消耗多少脑力值。显然可以预处理得到。
- 对于每一个状态，我们可以计算这个状态对第$q+1\dots m$类贡献的学分$d[q+1\dots m]$。
- $g[i][j]=\min\left\{g[i-1][j]+f[i]\left[k+s_i-d[i]\right] \middle| 0\le k\le T-\sum_1^m s_i\right\}$。总消耗要加上$h[z]$。

正解

- 求解f数组的时间复杂度：$O(N\times C)$。
- 求解h数组的时间复杂度：$O(p^2\times 2^p)$。
- 求解g数组的时间复杂度：$O(m\times(T-\sum_1^m s_i)^2+2^p\times p\times(T-\sum_1^m s_i)^2)$。
- 总时间复杂度在$1e8$的数量级，但该算法常数极小，且评测机速度极快，故**不存在**卡常的问题。