测试讲评 & 习题（二）

JAN,31TH,2023

朱宇博

测试（二）

- 取模
  - 2021 B卷day2T1
- 宝石
  - 2021 A卷day2T1
  - 2021 B卷day2T2
- 滚榜
  - 2021 A卷day2T2
  - 2021 B卷day2T3

测试讲评

P7521 [省选联考 2021 B 卷] 取模

给定 n 个正整数 a_i，请你在其中选出三个数 i,j,k （$i \neq j$，$i \neq k$，$j \neq k$），使得 $(a_i + a_j) \bmod a_k$ 的值最大。

solution

考虑固定模数 a_k。

令所有 $i \neq k$ 的 $a_i \bmod a_k$ 构成序列 b，那么答案最大只有两种情况：

1. $b_i + b_j \geq a_k$，那么因为 $b_i + b_j < 2a_k$，因此 $ans = b_i + b_j - a_k$。要使 ans 最大化，只需取最大的 b_i, b_j 即可；
2. $b_i + b_j < a_k$，直接对 b 排序跑一遍双指针即可得出最大的 ans。

此时仅考虑单一模数，时间复杂度O(nlogn)

solution

于是考虑减少做的次数。

首先对于相同的 a_k，答案肯定是一样的，只需做一遍。

其次，我们对于 a 按从大到小的顺序来做，如果当做到 a_i 时 $ans \geq a_i$，那就没必要做了。正确性显然。

- 设值域大小为V
- 时间复杂度O(n logn lonV)

令 a 已经从大到小排好序并去重。设 $a_1 > \cdots > a_x > ans \ge a_{x+1} > \cdots > a_n$。

因为 ans 是最大的答案，那么 $\forall 1 \le i \le x-2$，有

$$(a_{i+1} + a_{i+2}) \bmod a_i \le ans$$

显然

$$a_i > a_{i+1} > a_{i+2} > ans$$

那么

$$(a_{i+1} + a_{i+2}) \bmod a_i \le ans < a_{i+1} + a_{i+2} < 2a_i$$

所以

$$a_i \le a_{i+1} + a_{i+2} < 2a_i$$

于是把取模干掉，得到

$$a_{i+1} + a_{i+2} - a_i \le ans$$

变形得到

$$(a_{i+1} - ans) + (a_{i+2} - ans) \le (a_i - ans)$$

令 $f_i = a_i - ans$，那么

$$f_i \ge f_{i+1} + f_{i+2}$$

发现这东西长的很像倒着的 斐波那契数列。这玩意儿是呈指数级增长的。

又有 $f_1 = a_1 - ans \le a_1 \le V$，其中 V 代表值域，因此 f_i 的数量至多只有 $\log V$ 级别。

换句话说，我们只会扫至多 $O(\log V)$ 次就能找到最终答案。

P7518 [省选联考 2021 A/B 卷] 宝石

欧艾大陆上有 n 座城市，城市从 $1 \sim n$ 编号，所有城市经由 $n-1$ 条无向道路互相连通，即 n 座城市与 $n-1$ 条道路构成了一棵树。

每座城市的集市上都会出售宝石，总共有 m 种不同的宝石，用 $1 \sim m$ 编号。i 号城市的集市出售的是第 w_i 种宝石，一种宝石可能会在多座城市的集市出售。

K 神有一个宝石收集器。这个宝石收集器能按照顺序收集至多 c 颗宝石，其收集宝石的顺序为：$P_1, P_2, \ldots, P_c$。更具体地，收集器需要先放入第 P_1 种宝石，然后才能再放入第 P_2 种宝石，之后再能放入第 P_3 种宝石，以此类推。其中 $P_1, P_2, \ldots, P_c$ 互不相等。

K 神到达一个城市后，如果该城市的集市上出售的宝石种类和当前收集器中需要放入的种类相同，则他可以在该城市的集市上购买一颗宝石并放入宝石收集器中；否则他只会路过该城市什么都不做。

现在 K 神给了你 q 次询问，每次给出起点 s_i 与终点 t_i，他想知道如果从 s_i 号城市出发，沿最短路线走到 t_i 号城市后，他的收集器中最多能收集到几个宝石？（在每次询问中，收集器内初始时没有任何宝石。起点与终点城市集市上的宝石可以尝试被收集）

solution

我们考虑对给进来的 P_i 重标号一下，使之成为 $1\sim c$ 的序列，同时对每个 i 的 w 值也进行对应的重标

那么题意可以转化为求一条路径上 $u\to v$ 的最大的 cnt ，满足 $cnt\le c$ 且 $1\sim cnt$ 在这条路径上依次出现

显然在这条路径上，每个权值尽量早的出现最优

对于一条路径，可以拆分一条上链和一条下链

那么我们对于每个点 u 可以倍增预处理 $f_1[u][i]$ 表示在 u 到根的链上权值为 $w[u]+2^i$ 的节点最近在哪里

和 $f_2[u][i]$ 表示在 u 到根的链上权值为 $w[u]-2^i$ 的节点最近在哪里（为了处理下链）

对于上链可以直接求出权值为 1 的节点在哪里（因为必须从 1 开始起跳，怎么查找从 u 到根的链上权值为 x 的节点待会讲），并从 1 开始起跳，一直跳到深度不小于 lca 的节点

这样我们就得到了上链最大能匹配到哪里，不妨设为 x

那么我们可以在下链上二分答案，下界为 $x+1$ ，上界为 c，在 v 到根的链上查找权值为 mid 的点，并利用 f_2 不断往上跳，找到最大能匹配的后缀，看看是否能和前 $1\sim x$ 拼接在一起

（答案显然具有单调性的，因为如果权值较大的点能和上链拼接在一起，那么权值较小的点更加可以和上链拼在一起了）

至于二分的下界是 $x+1$ 的原因是 x 是在上链上出现的，不保证在下链上也存在 x

P7519 [省选联考 2021 A/B 卷] 滚榜

给定 n, m 和长度为 n 的数组 a_i 。问有多少种排列 p_n ，使得存在数组 b_i ，满足

- b_i 单调不减
- $\sum_{i=1}^{n} b_i = m$
- 依次对 a_{p_i} 加上 b_i ，满足操作后 a_{p_i} 为最大元素。

数据范围：$1 \leq n \leq 13, 1 \leq m \leq 500, 0 \leq a_i \leq 10^4$.

solution

看到 $n \leq 13$ 就考虑状压，于是随便想想就可以列出如下的状态定义：$f_{S,i,j,k}$ 表示考虑 S 集合，上一个人是 i，现在 $\sum b_i$ 已经用了 j，上一个 b_i 为 k 的方案数，直接暴力转移即可，时间复杂度为 $O(2^n n^2 m^2)$。

恭喜你，跑得比阶乘暴力还慢。

冷静看题。发现只用求排名的总方案数，而与 b_i 分配方式无关。可以得到的启发是寻找 b_i 的最佳分配方式。

对于某一确定的排列方式 $a_1, a_2, \cdots, a_n$，贪心地分配使得每个位置的 b_i 尽量小的方式易得：若当前的 a_i 大于 a_{i-1}，为了维护 b_i 不降，使得 $b_i = b_{i-1}$，否则 $b_i = b_{i-1} + a_{i-1} - a_i$。

根据这一结论，直接使用全排列可以获得 60 pts 的好成绩。

回到状压。考虑对贡献进行简单变形：$\sum b_i = \sum \max(a_{i-1} - a_i, 0)(n - i + 1)$，如此可以甩掉枚举上一个 b_i 的值这一步。设 $f(S, i, j)$ 表示前 $|S|$ 位为 S 内元素、第 $|S|$ 位为 i、已选总贡献为 j 的方案总数，在枚举状态的同时枚举下一个选的数，容易写出方程式。

最后统计 $f(U, i, j)$ 的和即可。

时间复杂度为 $\mathrm{O}(2^n n^2 m)$。可通过使用 lowbit 枚举元素等方式略微卡常。

上述方法忽略的细节是相等时的按编号排序，实现时应注意。

习题

CF587C Duff in the Army

有n个城市，由n-1条边连接。两个城市之间的道路是唯一的。 有m个人，住在这n个城市中，现在给出m个人，生活的城市编号。 你需要回答，从一个城市u到另一个城市v的路径中，编号前a小的人的编号是哪些?

- a<=10, n <= 1e5

思路1（树上倍增）

树上倍增，可以开一个$num[i][j]$,表示i这个点向前跳2^j-1格后到达的点，这样，对于两个点而言，就可以向上跳跃，直到交汇为止，答案则是归并后的ans数组。

此题的切入点主要在$a <= 10$这个条件，这样使我们每个vector只要保存最多10个元素就好了。

倍增这种数据结构，是树上问题求解的常用途径，遇到树的问题要优先考虑。

HDU 6031 Innumerable Ancestors

- 给两个集合A,B，求从两个集合中选出的点的LCA深度最大可以是多少

- 若x是u和v的lca，则x的父亲也必然是u和v的公共祖先，因此可以二分深度，
- 对集合A的每个节点求深度为mid的祖先，插入一个set，对于集合B也求出每个节点深度为mid的祖先，如果在set中，则当前二分的答案可行

P1084 [NOIP2012 提高组] 疫情控制

H 国有 n 个城市，这 n 个城市用 $n-1$ 条双向道路相互连通构成一棵树，1 号城市是首都，也是树中的根节点。

H 国的首都爆发了一种危害性极高的传染病。当局为了控制疫情，不让疫情扩散到边境城市（叶子节点所表示的城市），决定动用军队在一些城市建立检查点，使得从首都到边境城市的每一条路径上都至少有一个检查点，边境城市也可以建立检查点。但特别要注意的是，首都是不能建立检查点的。

现在，在 H 国的一些城市中已经驻扎有军队，且一个城市可以驻扎多个军队。一支军队可以在有道路连接的城市间移动，并在除首都以外的任意一个城市建立检查点，且只能在一个城市建立检查点。一支军队经过一条道路从一个城市移动到另一个城市所需要的时间等于道路的长度（单位：小时）。

请问最少需要多少个小时才能控制疫情。注意：不同的军队可以同时移动。

- 最大值最小化，二分答案
- 每个点都尽可能往上跳，封锁尽量大的子树
- 如果可以跳到1:
  - 记录到1的剩余时间，以及跳上来的儿子(先不封锁)
- 如果不能跳到1
  - 记录能跳上来的最浅节点，封锁该子树

- 按到达1的节点的剩余时间从小到大排序
- 如果其跳上来的子节点所在子树没有被控制，则回去控制

- 最后处理没有被控制的1的儿子节点，按时间排序，看仍在1号节点的军队是否能控制

P1600 [NOIP2016 提高组] 天天爱跑步

`小c` 同学认为跑步非常有趣，于是决定制作一款叫做《天天爱跑步》的游戏。《天天爱跑步》是一个养成类游戏，需要玩家每天按时上线,完成打卡任务。

这个游戏的地图可以看作一一棵包含 n 个结点和 $n-1$ 条边的树，每条边连接两个结点,且任意两个结点存在一条路径互相可达。树上结点编号为从 1 到 n 的连续正整数。

现在有 m 个玩家，第 i 个玩家的起点为 s_i，终点为 t_i。每天打卡任务开始时，所有玩家在第 0 秒同时从自己的起点出发，以每秒跑一条边的速度，不间断地沿着最短路径向着自己的终点跑去，跑到终点后该玩家就算完成了打卡任务。(由于地图是一棵树，所以每个人的路径是唯一的)

`小c` 想知道游戏的活跃度，所以在每个结点上都放置了一个观察员。在结点 j 的观察员会选择在第 w_j 秒观察玩家，一个玩家能被这个观察员观察到当且仅当该玩家在第 w_j 秒也正好到达了结点 j 。`小c` 想知道每个观察员会观察到多少人?

注意：我们认为一个玩家到达自己的终点后该玩家就会结束游戏，他不能等待一 段时间后再被观察员观察到。即对于把结点 j 作为终点的玩家：若他在第 w_j 秒前到达终点,则在结点 j 的观察员不能观察到该玩家；若他正好在第 w_j 秒到达终点,则在结点 j 的观察员可以观察到这个玩家。

- 测试点6–8: 链
  - 左 –> 右： $s = j-W_j$
  - 右 –> 左: $s = j+W_j$
  - 对于每个观察员为定值，开桶维护

- 测试点9-12: s_i为1
  - 观察到点当且仅当depth_j = W_j
  - 对于该观察员，能观察到的点为终点在其子树内的点
  - dfs统计信息即可

- 测试点13–16: $t\_i = 1$
  - 能观察到点当且仅当depth_j + W_j = depth_{s_i}
  - 对于该观察员，能观察到的点为起点在其子树内的点
  - 考虑差分，设bac[i]代表起点深度为i的终点个数，在遍历到当且节点前后的差值，即为子树内深度为i的起点个数

- 测试点17–20： 无限制
  - 将路径分为向上走和向下走两个阶段
  - 对于lca向上走的路径， 需要满足
    - depth_j+w_j = deepth_s
  - 对于lca向下走的路径， 需要满足
    - depth_j–w_j = depth_t – l + 1 (l为st路径长度)
  - 处理不合法与重复
  - 树上差分
    - s –> lca 在s处+ lca处–
    - lca –> t 在t处+ prt[lca]处减

CF626F Group Projects

有 n 个学生，每个学生有一个能力值 a_i。现在要把这些学生分成一些(任意数量的)组，每一组的“不和谐度”是该组能力值最大的学生与能力值最小的学生的能力值的差。求所有不和谐度之和不超过 k 的分组方案总数。

对于每一个数据，都有以下几种情况：

1.作为一组值的最小值

2.作为一组值的最大值

3.作为一组值的中间值

4.单独成为一组

而所有数据的最小值不可能作为一组值的最大值，所有数据的最大值反之。

考虑将数据升序排序，方便处理每一组值的最大/最小值。

设 $f[i][j][k]$ 表示前 i 个值（排序后），有 j 个组 **可以** 再选人（还没最大值），不和谐度为 k 的方案数。

根据上述四种情况，转移式为：

$$f[i][j][k] += f[i-1][j-1][k+a[i]]$$

（$k+a[i]$表示这一组减去最小值）

$$f[i][j][k] += f[i-1][j+1][k-a[i]] * (j+1)$$

（ $k - a[i]$ 表示这一组加上最大值，最大值可以是前面j组任意一组的最大值，所以结果乘 $j+1$ ）

$$f[i][j][k] += f[i-1][j][k] * j$$

（中间值可以在前面j组任意一组中，所以结果乘 j ）

$$f[i][j][k] += f[i-1][j][k]$$

（自开自闭~~简称自闭~~区间）

时，如果某组以2为最小值，8为最大值，它的k值变化应该是

$$-2.....+8=6$$

由于数据是排序的，所以也可以写成

$$+(3-2)..+(5-3)..+(7-5)..+(8-7)=1+2+2+1=6$$

它们是相同的。

~~自己推推就好了~~

所以k值在每次转移时，不必在最小值和最大值处变化，而是在每次转移时变化，变化值为 $a[i]-a[i-1]$

状态转移式变为：

$$f[i][j][k]+=f[i-1][j-1][k-(j-1)*(a[i]-a[i-1])]$$

$$f[i][j][k]+=f[i-1][j+1][k-(j+1)*(a[i]-a[i-1])]*(j+1)$$

$$f[i][j][k]+=f[i-1][j][k-j*(a[i]-a[i-1])]*j$$

$$f[i][j][k]+=f[i-1][j][k-j*(a[i]-a[i-1])]$$

需要注意的是：由于转移时最小值与最小值-1的差值并不需要加上，所以1、2转移要稍加变换。

最后只需要累加 $f[n][0][i]$ (i from 0 to k)即可。

而 $f[0][0][0]=1$

P7962 [NOIP2021] 方差

给定长度为 n 的非严格递增正整数数列 $1 \leq a_1 \leq a_2 \leq \cdots \leq a_n$。每次可以进行的操作是：任意选择一个正整数 $1 < i < n$，将 a_i 变为 $a_{i-1} + a_{i+1} - a_i$。求在若干次操作之后，该数列的方差最小值是多少。请输出最小值乘以 n^2 的结果。

其中方差的定义为：数列中每个数与平均值的差的平方的平均值。更形式化地说，方差的定义为 $D = \frac{1}{n}\sum_{i=1}^{n}(a_i - \bar{a})^2$，其中 $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$。

- 交换的本质：交换了差分数组上的两个数
- 因此：差分数组可以随便重排

- 结论：最优差分数列呈现单谷状

将差分数组排序，从小到大确定位置，容易发现只可能放在两端。

于是我们开始 DP。

拆分方差，

$$n^2 D = n\sum a_i^2 - (\sum a_i)^2$$

设 dp_i 表示当前确定的图像表示数组 $\sum a_i = i$ 时，$\sum a_i^2$ 的最小值，同时维护一个确定的值 s 表示已插入差分的和。

假设现在要插入一个 x。

x 如果插入在后，那么相当于在数组后跟上一个 $s+x$，有：

$$dp_{i+s+x} = dp_i + (s+x)^2$$

x 如果插入在前，相当于所有数 $+x$ 并在前跟上一个 0，有：

$$dp_{i+cnt\times x} = dp_i + cnt \times x^2 + 2 \times i \times x$$

两者都可以很好维护。

由于已插入数的数量 c 和 s 对于每个插入过程都是确定的，所以不用记录在状态里。

最后的答案就是 $\min(n \times dp_i - i^2)$