# DP选讲

仓鼠

# AGC034E

给你一棵$N$个点，标号为1到$N$的树，第$i$条边连接点$a_i$和$b_i$，还有一个长度为$N$的01串$S$，串$S$的第$i$个字符表示有多少枚棋子放在第$i$个节点。

Snuke想做下面的操作若干次:

- 选择两枚距离至少为2的棋子，将它们向中间移动一步。也就是说选择两个分别有至少一个棋子的节点$u$和$v$，考虑$u$到$v$的最短路径，这里要求最短路径至少有两条边。然后，将$u$上的一个棋子移动到路径上与它相邻的节点，并将$v$上的一个棋子移动到路径上与它相邻的节点。

通过重复这个操作，Snuke想把所有棋子移动到同一个节点。这能否做到？如果可以做到，找出做到这一点所需的最少操作次数。

$2 \leq N \leq 2000$

$|S| = N$

$S$只包含0和1，且至少有一个1。

$1 \leq a_i, b_i \leq N(a_i \neq b_i)$

边$(a_1, b_1)$，$(a_2, b_2)$，…，$(a_{N-1}, b_{N-1})$形成一棵树。

# AGC034E

- 枚举最后所有标记聚集到的点root。把这个点当成树根
- 可以发现，如果可行，那么一定存在一个过程，每次选择两个没有祖先子孙关系的标记，同时向上（向着它们的最近公共祖先处）移动1个距离，直到所有点都在root
- 这里就涉及到一个经典的模型：有若干个点被分成了若干个集合，每次要找两个在不同集合中的点匹配然后消掉。当$\mathrm{sum}-\max \geq \max$时，即$\mathrm{sum} \geq 2\times\max$，如果总共有偶数个点可以消完否则可以剩下恰好1个，消去了$\left\lfloor\frac{\mathrm{sum}}{2}\right\rfloor$对；否则还剩下$\max-(\mathrm{sum}-\max)$个来自最大集合的点，消去了$\mathrm{sum}-\max$对

# AGC034E

- 考虑现在在x点处做这个消除的过程。设F[x]表示x子树中最多消除多少对。显然可以构造方案，使得x子树中被消去在$[0, \mathrm{F}[\mathrm{x}]]$中的任意多对数。一个点u被拆成了$\mathrm{Dist}(\mathrm{u}, \mathrm{x})$个点。转移分情况讨论：
- $\mathrm{sum} - \max \geq \max$，那么$\mathrm{F}[\mathrm{x}] = \left\lfloor \frac{\mathrm{sum}}{2} \right\rfloor$
- $\mathrm{sum} - \max < \max$，那么需要最大子树u中的点自己来消，$\mathrm{F}[\mathrm{x}] = (\mathrm{sum} - \max) + \max\{\mathrm{k} | 0 \leq \mathrm{k} \leq \mathrm{F}[\mathrm{u}], 2 \times \mathrm{k} \leq 2 \times \max - \mathrm{sum}\}$
- root合法的充要条件是$\mathrm{F}[\mathrm{root}] = \frac{\sum_{\mathrm{u}} \mathrm{Dist}(\mathrm{u},\mathrm{root})}{2}$，同时这也是答案
- 有一个比这道题略难一点的题：清华集训2017榕树之心。理解了这道题之后可以去做一下

# CF908G

- 定义S(n)为将n在十进制下所有数位从小到大排序后得到的数
- 例如：$S(232) = 223$，$S(50394) = 3459$
- 给定X，求$\sum_{i=1}^{X} S(i)$对一个大质数取模的结果
- $1 \leq X \leq 10^{700}$

# CF908G

- 对于一个数字digit($1 \leq \mathrm{digit} \leq 9$)，其对答案的贡献可以表示成$\mathrm{c(digit)} = \sum 10^{\mathrm{i}}$的形式，就是其在每个数中对应位置贡献的和
- $\mathrm{Ans} = \sum_{\mathrm{digit}=1}^{9} \mathrm{digit} \times \mathrm{c(digit)} = \sum_{\mathrm{digit}=1}^{9} \sum_{\mathrm{i}=\mathrm{digit}}^{9} \mathrm{c(i)}$
- 考虑枚举digit如何计算$\sum_{\mathrm{i}=\mathrm{digit}}^{9} \mathrm{c(i)}$，实际上就取决于每个数中不小于digit的数位有多少个
- 直接做数位dp，记录不小于digit的数位有多少个即可

# AGC024F

给由长为不超过 $N$ 的 01 串组成集合 $S$ 和一个整数 $K$，求出最长的、是 $S$ 中 $K$ 个及以上字符串子序列的字符串，如果存在多个，输出字典序最小的。

$0 \le N \le 20$，$1 \le K \le |S|$，所有长度不超过 $N$ 的 01 串都可能同时出现在 $S$ 中。

# AGC024F

- 显然是考虑对于每个串$S$，计数它是多少个串的子序列
- 考虑构造出一个类似自动机的东西
- 设一个状态$(S, T)$表示现在有一个串$S$，它后面需要接一个$T$的子序列
- 转移就类似在子序列自动机上走的过程，在它后面添加$0$和$1$可以分别走到两个状态，还有一个转移是不加任何字符然后结束。这样状态就构成了一个DAG
- 同时这个DAG有个很好的性质，因为是用类似子序列自动机的方法构造出来的，所以两点之间路径唯一

# AGC024F

- 根据前面的分析，如果$S$是$T$的子序列，那么当且仅当$(\emptyset, T)$可以走到$(S, \emptyset)$，这里用$\emptyset$表示空串
- 因为路径唯一，所以直接在DAG上做路径计数即可
- 因为这个状态满足$|S| + |T| \leq N$，所以状态数是$O(N \times 2^N)$的。转移$O(1)$，时间复杂度也就是$O(N \times 2^N)$。
- 这道题的巧妙之处在于，考虑对于每个集合中的串$S$，让它对它的所有子序列不重不漏产生贡献。暴力只能是对于每个串在子序列自动机上走，但是如果注意到子序列自动机可以压下来，以及后面的一些相关性质，这道题就迎刃而解了

# CTT2017某位歌姬的故事

- 有一个长度为N的数列A，数列中每个数在$[1, A]$的范围内
- 有M个限制形如$(l, r, h)$表示$\max\limits_{l \le i \le r}\{A_i\} = h$
- 求可能的A的数量，对大质数取模
- $N \le 9 \times 10^8$
- $M \le 500$
- $A \le 9 \times 10^8$

# CTT2017某位歌姬的故事

- $\max\limits_{l\le i\le r}\{A_i\} = h$等价于$\forall l \le i \le r, A_i \le h$且$\exists l \le i \le r, A_i = h$
- 求出每个点的上限$up_i$。可以发现，$\exists l \le i \le r, A_i = h$的i只能由上限$up_i = h$的点来贡献
- 所以对于每个h的限制以及上限为h的点单独拿出来dp，满足每个限制区间中至少有一个点达到上限即可

# 笛卡尔树dp

- 笛卡尔树就是一颗treap，其中权值是下标，优先级是权值。显然排列和笛卡尔树是一一对应的，即每个排列都有唯一的笛卡尔树，每个大小为N的二叉树都有唯一的排列对应
- 笛卡尔树上有一些性质
- 以一个点$u$为根的子树就是以其为最小值的极长区间$[l, r]$。要么$rson(l-1)=u$，要么$lson(r+1)=u$
- 一些dp会以笛卡尔树为背景

# Myjnie

- Source：BZOJ4380
- 有N家洗车店从左往右排成一排
- 有M个人要来消费，第i个人会驶过第$a_i$个开始一直到第$b_i$个洗车店，且会选择这些店中最便宜的一个进行一次消费，但是如果这个最便宜的价格大于$c_i$，那么这个人就不洗车了
- 请给每家店指定一个价格，使得所有人花的钱的总和最大
- $N \leq 50$
- $M \leq 1000$

# Myjnie

- 首先将权值离散化，考虑区间 DP，设$F[i][j][k]$为区间$[i,j]$最小值为$k$时的最大收益（只考虑$i \le a \le b \le j$的人）
- 转移时枚举最小值所在位置$x$，那么可以用$F[i][x-1][p \ge k] + F[x+1][j][q \ge k] + cost(i,j,x,k)$来更新答案
- 时间复杂度是$O(N^3M)$
- 这个可以理解成把序列构成了一颗笛卡尔树，并且考虑在笛卡尔树上计算贡献。这样笛卡尔树dp的思路十分常见

# PERIODNI

- Source：BZOJ2616
- 有个棋盘，第i列高度为H[i]，底边平行
- 选择K个位置放置一个车，使得互相不能攻击。两个车能互相攻击，当且仅当在同一行或同一列，不能跨过空格子
- 计数方案数，对大质数取模
- $\mathrm{N}, \mathrm{K} \leq 500$
- $\forall 1 \leq \mathrm{i} \leq \mathrm{N}, \mathrm{H}[\mathrm{i}] \leq 10^6$

# PERIODNI

- 建出一颗笛卡尔树，实际上就是每次把最低的一层给消掉。这样也构成了一个树的结构
- 这样一个子树代表了一个完整的矩形
- 在树上做笛卡尔树dp，设$F[u][k]$表示子树$u$中放了$k$个车的方案数。转移首先把所有子树背包合并起来，然后如果要在当前矩形中放，就只能在剩下$Size[u]-k$个列中放，行的范围就是矩形的范围

# Histogram Coloring

- Source：AGC026D
- 给定一个$10^9$行$n$列的表格，从左往右、从上往下标号，并去掉一些格子，使得第$i$列只剩下下面的$h_i$行格子
- 要求对剩下的格子染成红蓝两种颜色，使得每个$2 \times 2$的格子里面有恰好2个红色格子、2个蓝色格子
- 求方案数，对大质数取模
- $n \le 100 \quad \forall 1 \le i \le n, h_i \le 10^9$

# Histogram Coloring

- 首先考虑一个完整的矩形如何计数答案
- 假如已经决定了前面的行，要新增一行。首先可以把上面一行全部翻转当成新增一行；特殊地，当上面一行是黑白相间的，可以直接把上面一行复制下来
- 对于原问题考虑笛卡尔树dp，实际上就是每次把最低的一层给消掉。这样也构成了一个树的结构，然后从下往上dp，每次记录是否黑白相间的情况
- 转移需要把所有上面的层合并，讨论一些情况

# 摩天大楼

- Source：LOJ#2743
- 将N个互不相同的整数$A_1, A_2, \cdots, A_N$任意排列成$B_1, B_2, \cdots, B_N$，要求$\sum_{i=1}^{N-1} |B_{i+1} - B_i| \le L$
- 计数方案数
- $N \le 100 \quad L \le 1000$

# 摩天大楼

- 考虑一个把所有元素按照从大到小的顺序放到排列中正确的位置的过程。那么每次插入一个元素，其所在的连续段就是以其为根在笛卡尔树上的子树
- 把原序列排序后做个差分，那么在这个过程中，每个时刻序列都形成了若干个连续段。一个连续段的端点和其相邻的还未被填上的数的差值会被拉大，拉大的数值就是当前差分的值
- 所以可以记录形成的连续段的情况。设$F[i][s][x][a \in \{0,1\}][b \in \{0,1\}]$表示考虑到第$i$个元素，当前差值的和为$s$，连续段的个数为$x$，序列的左右端点有没有被加入。可以根据这些信息算出，被拉大差值的两个相邻元素的数目。每次加入元素分情况讨论：新建段、紧贴着一个段、合并了两个段

# Tree Depth

- Source：USACO2019.12Platinum LOJ#3228
- 考虑所有长度为N有恰好K个逆序对的排列
- 对于每个位置i，求出其在所有满足条件的排列中，在笛卡尔树上的深度和
- 对读入的质数MOD取模
- $\mathrm{N} \leq 300$
- $\mathrm{K} \leq \frac{\mathrm{N}(\mathrm{N}-1)}{2}$
- $\mathrm{MOD} \in [10^8, 10^9+9]$

# Tree Depth

- j是i在笛卡尔树上祖先的充要条件是，$\min_{\min\{i,j\} \le k \le \max\{i,j\}} \{P_k\} = P_j$
- 考虑一个按照从小到大的顺序把所有数插入的过程。那么插入x的时候，它可以通过选择所在的位置使得新增的逆序对个数为$[0, x-1]$中的任何一个数
- 或者考虑一个从前往后决定元素的过程。那么可以通过选择第i个位置在前i个位置中的相对大小，使得新增的逆序对个数为$[0, i-1]$中的任何一个数
- 这里我们采用第二个过程来考虑这道题

# Tree Depth

- 枚举j是i在笛卡尔树上祖先。考虑把插入的过程分成两部分：第一部分从i开始向j的方向，决定每个元素在当前排列中的相对位置；第二部分从i开始向j的反方向，决定每个元素在当前排列中的相对位置
- 这样，每次新产生的逆序对数目区间也是$[0,1],[0,2],\cdots,[0,N-1]$这样。除了j需要成为祖先当$j<i$时，必定产生0个逆序对，当$j>i$时，必定产生$j-i$个逆序对
- 先假装下标为i的每次产生逆序对个数在$[0,i-1]$内的贡献，做一次dp。每次枚举j，撤销掉其本来的贡献；然后枚举i计算其对i答案的贡献
- 时间复杂度$O(N^3)$

# dp套dp

- dp套dp是给定一个DP问题A，用另一个dp去计算一种可能的A的输入，使得A的dp结果为x
- 说白了就是，外层的dp的状态是另一个dp的结果
- 这样的问题，往往需要深入挖掘内层dp的性质，有时候还要对状态数有一个~~合理的估计~~甚至是大胆的猜想

# BZOJ3864

在本题中字符集大小为 $4$。

给定一个长度为 $n$ 的字符串 $S$，对于每个 $0 \le k \le n$，计算有多少长度为 $m$ 的字符串 $T$ 满足 $LCS(S, T) = k$。

- $n \le 15$。
- $m \le 1000$。

# BZOJ3864

- LCS的dp相信大家都很熟练了。设$f[i][j]$表示T的前i位和S的前j位的LCS是多少
- 设$F[i][S]$表示考虑T的前i位，dp数组$f[i]$是S的方案数
- 注意到$f[i][j-1] \le f[i][j] \le f[i][j-1]+1$，所以S中只要记录相邻两项的差值，不是0就是1
- 转移枚举字符是什么，可以用预处理转移做到$O(1)$
- 时间复杂度是$O(M \times 2^N)$

# LOJ6274

NiroBC 姐姐脑洞了两个数字 $x$ 和 $y$，它们满足 $x \vee y = T$，且 $L_x \le x \le R_x, L_y \le y \le R_y$， NiroBC 姐姐想知道 $x \wedge y$ 有多少种不同的取值，若有多组 $(x, y)$ 的 $x \wedge y$ 值相同，则只算一次。

（其中 $\vee$ 表示按位取或，**`C/C++`**中写作**`|`**，**`Pascal`**中写作**`or`**）

（其中 $\wedge$ 表示按位取与，**`C/C++`**中写作**`&`**，**`Pascal`**中写作**`and`**）

对于所有数据，$0 \le T, L_x, R_x, L_y, R_y < 2^{60}$，$L_x \le R_x$，$L_y \le R_y$。

# LOJ6274

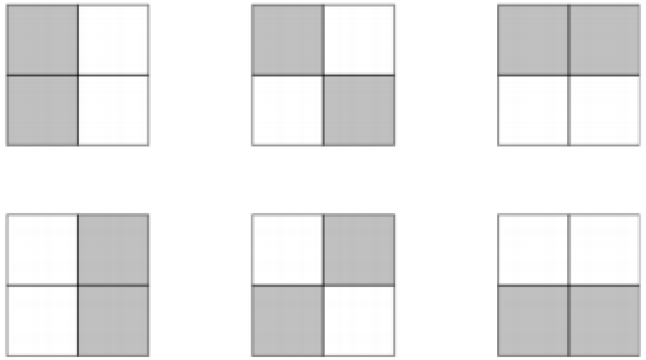
- 假设知道$\mathrm{x\ or\ y}=\mathrm{T}$且$\mathrm{x\ and\ y}=\mathrm{V}$，判断是否可行，可以做一个数位dp
- f[i][a][b][c][d]表示考虑到第i位，x与Lx和Rx的关系，y与Ly和Ry的关系，这种情况下是否可行
- 然后写一个dp，把所有dp结果压起来就可以了
- F[i][S]表示dp到第i位，dp状态为S的V有多少个
- 转移枚举V的第i位是什么

# 插头dp

- 实际上就是一种特殊一点的状压dp
- 把比较基础的按行/按列转移变成按格转移，这时候状态记录的就是一个“轮廓线”上的状态。这个轮廓线就是一个Z字型

# Domino Colorings

请统计有多少 $n \times m$ 的黑白棋盘可以被 $1 \times 2$ 的黑白相间的骨牌精确覆盖。



- $n \leq 6$。
- $m \leq 300$。

# Domino Colorings

- 若已经知道了每个格子的颜色，那么可以dp判断是否能由骨牌铺成。具体而言，设$f[S]$表示当前轮廓线上的点是否被匹配的状态为$S$时，能否由骨牌铺成。这里每次转移只会涉及到轮廓线上的点
- 现在要做一个dp求解原问题的答案，所以要把上面的插头dp压起来。设$F[i][j][c][V]$表示用插头dp的顺序逐格转移到$(i,j)$，轮廓线上的格子颜色为$c$，每个dp的结果为$V$的方案数
- 打个表发现状态数不到$2000$

# 决策单调性

- 常见的优化方法：分治和决策栈
- 分治：$\mathrm{solve}(\mathrm{l},\mathrm{r},\mathrm{sl},\mathrm{sr})$表示$[\mathrm{l},\mathrm{r}]$中的最优决策点的范围是$[\mathrm{sl},\mathrm{sr}]$。算出$\mathrm{mid}=\left\lceil\frac{\mathrm{l}+\mathrm{r}}{2}\right\rceil$的最优决策smid，然后分治下去做$\mathrm{solve}(\mathrm{l},\mathrm{mid}-1,\mathrm{sl},\mathrm{smid})$和$\mathrm{solve}(\mathrm{mid}+1,\mathrm{r},\mathrm{smid},\mathrm{sr})$
- 决策队列：一边做一边维护之后所有点的决策。每次在决策队列的前缀或者后缀处二分
- 在dp不是分阶段的时候或者有些其他限制的时候，可能没法分治

# CF868F

- 给定$A_1, A_2, \cdots, A_N$和K，把A分成K个连续段，使得每一段中重复对个数之和最小
- $N \leq 10^5$
- $K \leq 20$

# CF868F

- $F[k][i]$表示把前i个元素分成k段的最优解
- 显然分层做，$F[k]$从$F[k-1]$转移过来。有决策单调性
- 主要问题在于如何快速计算一个区间的重复对个数。可以直接像莫队一样做。这与正常的枚举复杂度是同阶的，因此每一层分治的复杂度仍然是$O(N\log_2 N)$
- 总的时间复杂度是$O(KN\log_2 N)$

# USACO2019FEB Mowing Mischief

给定平面上的一些点，求这些点的一个$LIS$，并且还需要满足下列式子最小：

$$\sum_{i=1}^{n-1}(a[i+1].x-a[i].x)*(a[i+1].y-a[i].y)$$

- LIS指的是两维坐标都递增
- $N \le 200000$

# USACO2019FEB Mowing Mischief

- 先正常做一遍LIS，然后按照LIS的dp值分层。这样每一层都是一条反链，即横坐标递增、纵坐标递减
- 下面考虑最小化题目给的式子
- 按照$\mathrm{LIS_i}$分层dp，$\mathrm{F[i]} = \min_{\mathrm{X_j \le X_i, Y_j \le Y_i, LIS_j = LIS_i - 1}} \{\mathrm{F[j]} + (\mathrm{X_i} - \mathrm{X_j})(\mathrm{Y_i} - \mathrm{Y_j})\}$
- 假如不考虑两维都要小于等于的限制，可以证明，对于两个决策a和b满足$a < b$，一定存在一个分界点，在这之前b更优、之后a更优
- 考虑上限制意味着每个点的决策有一个区间的限制。用线段树优化，每个节点对于该区间中的决策维护一个决策队列

# CF1110H

You are given two integers $l$ and $r$.

Let's call an integer $x$ *modest*, if $l \leq x \leq r$.

Find a string of length $n$, consisting of digits, which has the largest possible number of substrings, which make a modest integer. Substring having leading zeros are not counted. If there are many answers, find lexicographically smallest one.

If some number occurs multiple times as a substring, then in the counting of the number of modest substrings it is counted multiple times as well.

## Input

The first line contains one integer $l$ ($1 \leq l \leq 10^{800}$).

The second line contains one integer $r$ ($l \leq r \leq 10^{800}$).

The third line contains one integer $n$ ($1 \leq n \leq 2\,000$).

## Output

In the first line, print the maximum possible number of modest substrings.

In the second line, print a string of length $n$ having exactly that number of modest substrings.

If there are multiple such strings, print the lexicographically smallest of them.

# CF1110H

- 考虑如何描述$x \leq R$这个条件
- x的位数小于R的位数时，一定满足。下面只考虑位数相等的情况
- 枚举x和R的最长公共前缀，那么下一位一定是满足小于的关系，于是再枚举下一位是什么。这样就得到了一个x的前缀
- 实际上，上面都可以描述成， x匹配到了某一个前缀，同时位数不小于某一个下限，就可以说明$x \leq R$

# CF1110H

- 把所有匹配到的限制插入到AC自动机中
- 不难发现，所有限制的总长是$O(\mathrm{len} \times \mathrm{SIGMA})$的
- 然后直接在AC自动机上做数位dp即可

# AGC022E

定义一个长度为奇数 $N$ 的 01 串是美丽的，当且仅当每次将连续三个位替换为它们的中位数 $\frac{N-1}{2}$ 次后，这个串变成一个字符 `1`.

现在输入了一个字符串 $S$，由 `0`, `1` 和 `?` 组成，求将问号替换为 `0` 和 `1` 得到一个美丽的字符串的方案数。对 $10^9+7$ 取模。

$|S| \leq 3 \times 10^5$ 且 $|S|$ 是奇数。

# AGC022E

- 很多这样的题，套路就是先考虑一个判定的过程，然后把过程写到dp里面去就可以了。所有我们先考虑判定的过程
- 维护一个栈，这个栈满足存在一个分界点，前面全是1后面全是0。这里前面后面指的是，把栈从栈底向栈顶写下来。然后从左往右扫
- 如果当前是0，就把0压入到栈里面去。如果有连续的三个0，可以把这3个0消掉2个变成1个0
- 如果当前是1，当栈顶是一个0的时候显然可以相互抵消（因为再找一个数取中位数，取决于再找的数）。否则直接压入栈

# AGC022E

- 一个显然的事情是，栈里面0的个数不超过2。同时，当栈里面1的个数达到2时，那么最终一定可以让字符串变成1。因此，当1的个数超过2时，可以看成就是2
- 所以栈的种类数只有$3 \times 3 = 9$种
- 最终可以变成1的条件，就是最后的栈里面1个数不少于0个数
- 直接把当前栈长成什么样记下来，从左往右dp即可
- 时间复杂度是$O(N)$

谢谢大家！
祝大家省选顺利！