2013 Multi-University Training Contest 10

ftiasch

August 22, 2013

Sum

- $$S(k) = \binom{N-1}{k-1} \implies \sum_{k=1}^{N} S(k) = \sum_{k=1}^{N} \binom{N-1}{k-1} = 2^{N-1}$$
- **组合证明** $S(k)$ 是 “把 $(k-1)$ 个木板放进 $(N-1)$ 个间隔的方案数”；所以 $\sum_{k=1}^{N} S(k)$ 是 “把木板放进 $(N-1)$ 个间隔的方案数”

- $2^N \bmod (10^9+7)$ for $N = 10^{100000}$?

- $2^N \bmod (10^9+7)$ for $N = 10^{100000}$?
- 由费马小定理，

$$2^N \equiv 2^{N \bmod (10^9+6)} \pmod{10^9+7}$$

Sum

时间复杂度 $O(\log N + \log\log N)$

Y

- Why "Y"?

Y

- Why “Y”?
- 长得像:)

- 枚举 “Y” 的中心 u
- A, B, C 分别属于 3 棵**不同**的子树

- 枚举 “Y” 的中心 u
- A, B, C 分别属于 3 棵**不同**的子树
- 假设 u 的子树大小分别是 $s_1, s_2, \ldots, s_m$
- 方案数是

$$\sum_{1 \le i < j < k \le m} s_i \cdot s_j \cdot s_k = \frac{(\sum s_i)^3 - 3(\sum s_i)(\sum s_i^2) + 2\sum s_i^3}{6}$$

- 枚举 “Y” 的中心 u
- A, B, C 分别属于 3 棵**不同**的子树
- 假设 u 的子树大小分别是 $s_1, s_2, \ldots, s_m$
- 方案数是

$$\sum_{1 \le i < j < k \le m} s_i \cdot s_j \cdot s_k = \frac{(\sum s_i)^3 - 3(\sum s_i)(\sum s_i^2) + 2 \sum s_i^3}{6}$$

- 有点麻烦:(

- 让我们愉快地补集转换吧
- 假设路径 P **按顺序**经过 A, B, C 且 $A < C$，枚举 B
- A, C 分别属于 2 棵**不同**的子树

- 让我们愉快地补集转换吧
- 假设路径 P **按顺序**经过 A, B, C 且 $A < C$，枚举 B
- A, C 分别属于 2 棵**不同**的子树
- 假设 B 的子树大小分别是 $s_1, s_2, \ldots, s_m$
- 方案数是

$$\sum_{1 \leq i < j \leq m} s_i \cdot s_j = \frac{(\sum s_i)^2 - \sum s_i^2}{2}$$

时间复杂度 $O(N)$

Answer

- **世界观** （点）权是 1 或 2 的环套内向树，判断长度是 d 的路径是否存在

Answer

- **世界观** （点）权是 1 或 2 的环套内向树，判断长度是 d 的路径是否存在
- 对于 $d > 0$，当且仅当

  $$(C_1 = C_2 = \cdots = C_N = 2) \wedge (d \equiv 1 \pmod 2)$$

  无解
- 证明？

Answer

- **世界观** （点）权是 1 或 2 的环套内向树，判断长度是 d 的路径是否存在
- 对于 $d > 0$，当且仅当

  $$(C_1 = C_2 = \cdots = C_N = 2) \land (d \equiv 1 \pmod 2)$$

  无解
- 证明？
- 若有 1，从 1 出发不断扩展直到 $\geq d$。如果是 $d+1$ 则去掉开始的 1

时间复杂度 $O(N+Q)$

Editor

- 栈 A, B维护光标之前（之后）的序列

  ```
  I x A.push(x)
    D A.pop()
    L B.push(A.pop())
    R A.push(B.pop())
  ```

- 对于栈 A 中的每个元素，额外维护前缀和，以及前缀和的最大值

Editor

- 栈 `A`, `B`维护光标之前（之后）的序列

  `I` x `A.push(x)`
  `D` `A.pop()`
  `L` `B.push(A.pop())`
  `R` `A.push(B.pop())`

- 对于栈 A 中的每个元素，额外维护前缀和，以及前缀和的最大值
- （我猜）Splay Tree 会退化成上述结构，大概可以通过

时间复杂度 $O(Q)$

Flow

- 如果解是**树**?

Flow

- 如果解是树？
- 递归构造
  - 令 $f^* = \min_{a \neq b} F_{a,b}$
  - 令 $A = \{v_0\} \cup \{v : F_{v,v_0} > f^*\}, B = \{v : F_{v,v_0} = f^*\}$。
    - 若 $B = \emptyset$，失败
    - 若 $\exists (a \in A, b \in B) F_{a,b} > f^*$，失败
  - 递归构造 A, B
  - 取 $a^* \in A, b^* \in B$，加边 (a^*, b^*)，容量是 f^*

Flow

- 如果解是树？
- 递归构造
  - 令 $f^* = \min_{a \neq b} F_{a,b}$
  - 令 $A = \{v_0\} \cup \{v : F_{v,v_0} > f^*\}, B = \{v : F_{v,v_0} = f^*\}$。
    - 若 $B = \emptyset$，失败
    - 若 $\exists (a \in A, b \in B) F_{a,b} > f^*$，失败
  - 递归构造 A, B
  - 取 $a^* \in A, b^* \in B$，加边 (a^*, b^*)，容量是 f^*
- 一定存在解是树 $\implies$ *Gomory–Hu tree*

时间复杂度 $O(N^3)$

Game

- 设 $win(i, x, y)$ 表示当前可以买的物品是 i，先手有 x 元，后手有 y 元时，先手是否必胜
- $$win(i, x, y) \iff \exists j((j > i) \wedge (x \geq s_i - s_j) \wedge \neg win(j, y, x - s_i + s_j))$$
- 其中 $s_i = C_i + C_{i+1} + \cdots + C_N$

Game

- 设 $win(i,x,y)$ 表示当前可以买的物品是 i，先手有 x 元，后手有 y 元时，先手是否必胜
- $$win(i,x,y) \iff \exists j((j>i) \wedge (x \geq s_i - s_j) \wedge \neg win(j,y,x-s_i+s_j))$$
- 其中 $s_i = C_i + C_{i+1} + \cdots + C_N$
- 注意到 $x+y = A+B-s_1+s_i$，即 $win(i,x) := win(i,x,y)$

Game

- $win(i, x) \implies win(i, x+1)$
- 设 $m(i) = \min\{x : win(i, x)\}$，则 $\neg win(i, x) \iff x \leq m(i) - 1$

Game

- $win(i,x) \implies win(i,x+1)$
- 设 $m(i)=\min\{x: win(i,x)\}$，则 $\neg win(i,x) \iff x \le m(i)-1$
- 令 $D = A+B-s_1+s_i$
- $$\begin{aligned} & m(i) \\ =& \min\{x : \exists j((j>i) \wedge (x \ge s_i - s_j) \wedge \neg win(j, D-x))\} \\ =& \min\{x : \exists j((j>i) \wedge (x \ge s_i - s_j) \wedge D - x \le m(j)-1\} \\ =& \min\{\max\{s_i - s_j, D - m(j)+1\} : j > i\} \\ =& \min\{\max\{s_i - s_j, A+B-s_1+s_i-m(j)++1\} : j>i\} \\ =& s_i + \min\{\max\{-s_j, A+B-s_1-m(j)+1\} : j>i\} \end{aligned}$$
- 只要测试 $A \ge m(1)$

时间复杂度 $O(N)$

Count

- **世界观** $10^9 \times 10^9$ 的点阵，有 K 个黑点，求至少含有 1 个黑点的矩形数量
- $\iff$ 不含黑点的矩形数量

Count

- **世界观** $10^9 \times 10^9$ 的点阵，有 K 个黑点，求至少含有 1 个黑点的矩形数量
- $\Longleftrightarrow$ 不含黑点的矩形数量
- $10^3 \times 10^3$?
- 枚举矩形的下底，设 h_i 表示第 i 列最多向上扩展（不包含黑点）的长度，共有

  $$\sum_{i=1}^{N}\sum_{j=i}^{N}\min\{h_i, h_{i+1}, \ldots, h_j\}$$

  种
- 单调队列 $O(N)$

- $10^9 \times 10^3$?
- 注意到

$$\sum_{i=1}^{N}\sum_{j=i}^{N}\min\{h_i, h_{i+1}, \ldots, h_j\} = \sum_{i=1}^{N} h_i \cdot k_i$$

，其中 k_i 由 $\{h_1, h_2, \ldots, h_N\}$ 的**相对大小关系**决定

- 而对于空行，$h_i' = h_i + 1$，$\{h_i\}$ 和 $\{h_i'\}$ 的相对关系相同，只需计算出 $\{k_i\}$ 后一起处理

- $10^9 \times 10^3$?
- 注意到

$$\sum_{i=1}^{N}\sum_{j=i}^{N}\min\{h_i, h_{i+1}, \ldots, h_j\} = \sum_{i=1}^{N} h_i \cdot k_i$$

，其中 k_i 由 $\{h_1, h_2, \ldots, h_N\}$ 的**相对大小关系**决定

- 而对于空行，$h'_i = h_i + 1$，$\{h_i\}$ 和 $\{h'_i\}$ 的相对关系相同，只需计算出 $\{k_i\}$ 后一起处理
- $10^9 \times 10^9$?
- 对列进行离散化

时间复杂度 $O(K^2)$

Convex

- 仅当 3 点共线时，凸包形状（可能）发生变化
- 若点 i, j, k 共线，即

  $$(P_i(t) - P_j(t)) \times (P_i(t) - P_k(t)) = 0$$

  是关于 t 的 2 次方程，至多 2 解
- 事件点 $O(N^3)$

Convex

- 仅当 3 点共线时，凸包形状（可能）发生变化
- 若点 i, j, k 共线，即

$$(P_i(t) - P_j(t)) \times (P_i(t) - P_k(t)) = 0$$

  是关于 t 的 2 次方程，至多 2 解
- 事件点 $O(N^3)$
- 若 $t \in [a, b]$ 时，凸包由点 $i_1, i_2, \ldots, i_k$ 组成，则

$$S(t) = \frac{1}{2} \sum_{1 \le j \le k} P_{i_j}(t) \times P_{i_{(j+1) \bmod k}}(t)$$

  是关于 t 的 2 次函数，积分是 3 次函数

时间复杂度 $O(N^4 \log N)$

Rectangle

- **世界观** $10^9 \times 10^9$ 的点阵，初始点权都是 1，对矩形 R_i 的点：
  - 求和，记为 v_i
  - 权值增加 v_i
- 方案总数 $\sum_{i=1}^{N} v_i \cdot (X_{i2} - X_{i1} + 1) \cdot (Y_{i2} - Y_{i1} + 1)$

- 分治操作，考虑前一半增加的权值对后一半求和的贡献
- 类似矩形面积并，离线线段树维护

时间复杂度 $O(N\log^2 N)$