试题讲评

karin0 Irides

nwnusch

2019 年 8 月 2 日

phantasm

求 1, 2, ..., n 有多少个长为 m 的子序列 a，满足

- $a_1 = 1, a_m = n$
- $\forall i, a_{i+1} - a_i \geq k$

保证这样的子序列存在。只需判断方案数的奇偶性。数据有 T 组。

$$n, m, k \leq 10^9, T \leq 2 \times 10^6.$$

30 分的算法

枚举集合/搜索。

复杂度：$O(T \cdot 2^n)$。

不加说明时，这里复杂度中的 n, m, k 分别指各组询问中最大的 n, m, k。

对 $m \le 3$ 的算法

$m = 2$ 时，方案数就是 1。

对 $m \le 3$ 的算法

$m=2$ 时，方案数就是 1。

$m=3$ 时，a_2 不能选最前和最后的 $k+1$ 个元素，所以方案数就是 $n-2(k+1)$。

对 $m \le 3$ 的算法

$m = 2$ 时，方案数就是 1。

$m = 3$ 时，a_2 不能选最前和最后的 $k + 1$ 个元素，所以方案数就是 $n - 2(k + 1)$。

复杂度：$O(T)$。

综合之前的算法，可以获得 35 分。

对 $k = 1$ 的算法

没有第二条约束时，答案就是 $\binom{n-2}{m-2}$，可以 $O(nm)$ 递推出组合数来回答。

复杂度：$O(T + nm)$。

综合之前的算法，可以获得 40 分。

对 $k \le 2$ 的算法

$k = 2$ 时，第二条约束为子序列中相邻的元素不是相邻整数。

对 $k \leq 2$ 的算法

$k = 2$ 时，第二条约束为子序列中相邻的元素不是相邻整数。

推导一下或者找规律可以发现答案是 $\binom{n-m-1}{m-2}$。

复杂度：$O(T + nm)$。

综合之前的算法，可以获得 45 分。

动态规划的算法

由小 A 移动的过程可以想到一类序列 dp 的过程。

动态规划的算法

由小 A 移动的过程可以想到一类序列 dp 的过程。

设 $f(i, j)$ 表示 $a_i = j$ 时子序列 a 前 i 个位置可能情况的数量，即小 A 向东跳时经过 i 个位置到达 j 的方案数。

动态规划的算法

由小 A 移动的过程可以想到一类序列 dp 的过程。

设 $f(i,j)$ 表示 $a_i = j$ 时子序列 a 前 i 个位置可能情况的数量，即小 A 向东跳时经过 i 个位置到达 j 的方案数。

$$f(i,j) = \sum_{p=1}^{j-k} f(i-1,p)$$

动态规划的算法

由小 A 移动的过程可以想到一类序列 dp 的过程。

设 $f(i,j)$ 表示 $a_i=j$ 时子序列 a 前 i 个位置可能情况的数量，即小 A 向东跳时经过 i 个位置到达 j 的方案数。

$$f(i,j)=\sum_{p=1}^{j-k} f(i-1,p)$$

转移与 k 有关，可以对每组询问都做一遍 dp。

动态规划的算法

由小 A 移动的过程可以想到一类序列 dp 的过程。

设 $f(i,j)$ 表示 $a_i = j$ 时子序列 a 前 i 个位置可能情况的数量，即小 A 向东跳时经过 i 个位置到达 j 的方案数。

$$f(i,j) = \sum_{p=1}^{j-k} f(i-1,p)$$

转移与 k 有关，可以对每组询问都做一遍 dp。

维护前缀和优化转移，复杂度是 $O(Tnm)$。

可以获得 55 分。

动态规划的算法

$k \leq 200$ 时，不同的 k 不会超过 200 个，所以最多只需要做 200 遍 dp。

动态规划的算法

$k \leq 200$ 时，不同的 k 不会超过 200 个，所以最多只需要做 200 遍 dp。

复杂度：$O(T + nmk)$。

可以获得 65 分。

90 分算法

推式子。

注意到第二条约束与 a 的差分序列有关，考虑统计差分序列 $b_i = a_{i+1} - a_i$。序列 b 需要满足

- $\forall i, b_i$ 是 $[k, n]$ 上的整数
- $\sum_{i=1}^{m-1} b_i = n - 1$

90 分算法

推式子。

注意到第二条约束与 a 的差分序列有关，考虑统计差分序列 $b_i = a_{i+1} - a_i$。序列 b 需要满足

- $\forall i, b_i$ 是 $[k, n]$ 上的整数
- $\sum_{i=1}^{m-1} b_i = n - 1$

由于 $a_1 = 1$ 已确定，可以发现所有满足以上约束的长为 $m - 1$ 的序列 b 与原来的序列 a 一一对应。

90 分算法

有限项的正整数序列的和确定时，其方案数可以用隔板法计算，所以构造 $c_i = b_i - (k-1)$ 来去除第一条约束，序列 c 满足

- $\forall i, c_i$ 是正整数
- $\sum_{i=1}^{m-1} c_i = n - 1 - (m-1)(k-1)$

90 分算法

有限项的正整数序列的和确定时，其方案数可以用隔板法计算，所以构造 $c_i = b_i - (k-1)$ 来去除第一条约束，序列 c 满足

- $\forall i, c_i$ 是正整数
- $\sum_{i=1}^{m-1} c_i = n - 1 - (m-1)(k-1)$

满足约束的长为 $m-1$ 的序列 c 与序列 b 也是一一对应的。

90 分算法

有限项的正整数序列的和确定时，其方案数可以用隔板法计算，所以构造 $c_i = b_i - (k-1)$ 来去除第一条约束，序列 c 满足

- $\forall i, c_i$ 是正整数
- $\sum_{i=1}^{m-1} c_i = n - 1 - (m-1)(k-1)$

满足约束的长为 $m-1$ 的序列 c 与序列 b 也是一一对应的。

由隔板法即得其方案数为

$$\binom{n-2-(m-1)(k-1)}{m-2}$$

复杂度：$O(T + nm)$。

100 分算法

由 Lucas 定理，$\binom{n}{k} \equiv 1 \pmod 2$ 当且仅当二进制下 k 的各位都不大于 n 的对应位，即 n and $k = k$，其中 and 为二进制按位与。

复杂度：$O(T)$。

skylines

给一个 n 个点的树，点有权 c_i，边也有权。有 q 次询问，每次给定 u，询问

$$f(u) = \min_{v \neq u}(\text{dist}(u, v) + c_u - c_v)$$

$n, q \leq 2 \times 10^5$.

$O(qn \log n)$

对于每次询问，枚举所有点计算答案，其中通过求 lca 来求树上距离。

$O(qn \log n)$

对于每次询问，枚举所有点计算答案，其中通过求 lca 来求树上距离。

复杂度：$O(qn \log n)$。

$O(qn\log n)$

对于每次询问，枚举所有点计算答案，其中通过求 lca 来求树上距离。

复杂度：$O(qn\log n)$。

可以获得不超过 60 分。

$O(qn)$

对于每次询问，以 u 为根 dfs 一遍整棵树，同时统计答案。

$O(qn)$

对于每次询问，以 u 为根 dfs 一遍整棵树，同时统计答案。

复杂度：$O(qn)$。

可以获得 60 分。

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

$$f(i) = \min_{j \neq i}(\mathrm{dist}(i, j) + c_i - c_j)$$

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

$$\begin{aligned} f(i) &= \min_{j \neq i}(\text{dist}(i, j) + c_i - c_j) \\ &= \min\{\min_{j<i}(a_i + c_i - a_j - c_j), \min_{j>i}(-a_i + c_i + a_j - c_j)\} \end{aligned}$$

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

$$
\begin{aligned}
f(i) &= \min_{j \neq i}(\operatorname{dist}(i,j) + c_i - c_j) \\
&= \min\{\min_{j<i}(a_i + c_i - a_j - c_j), \min_{j>i}(-a_i + c_i + a_j - c_j)\} \\
&= \min\{a_i + c_i + \min_{j<i}(-a_j - c_j), -a_i + c_i + \min_{j>i}(a_j - c_j)\}
\end{aligned}
$$

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

$$\begin{aligned} f(i) &= \min_{j \neq i}(\operatorname{dist}(i,j) + c_i - c_j) \\ &= \min\{\min_{j<i}(a_i + c_i - a_j - c_j), \min_{j>i}(-a_i + c_i + a_j - c_j)\} \\ &= \min\{a_i + c_i + \min_{j<i}(-a_j - c_j), -a_i + c_i + \min_{j>i}(a_j - c_j)\} \end{aligned}$$

遍历两次，分别求出每个 i 左边最小的 $-a_j - c_j$ 和右边最小的 $a_j - c_j$ 即可处理出所有 $f(i)$。

链上的算法

树的形态为一条链 $1, 2, \cdots, n$ 时，待求式可以进一步化简。

设 1 到 i 的距离为 a_i，则

$$
\begin{aligned}
f(i) &= \min_{j \neq i}(\mathrm{dist}(i,j) + c_i - c_j) \\
&= \min\{\min_{j<i}(a_i + c_i - a_j - c_j), \min_{j>i}(-a_i + c_i + a_j - c_j)\} \\
&= \min\{a_i + c_i + \min_{j<i}(-a_j - c_j), -a_i + c_i + \min_{j>i}(a_j - c_j)\}
\end{aligned}
$$

遍历两次，分别求出每个 i 左边最小的 $-a_j - c_j$ 和右边最小的 $a_j - c_j$ 即可处理出所有 $f(i)$。

复杂度为 $O(n)$，可以获得 40 分。综合之前的算法，可以获得 80 分。

100 分算法

对于有根树，一个点所选择的对应点仅在其子树内的情形，答案可以通过树形 dp 递推得到。

100 分算法

对于有根树，一个点所选择的对应点仅在其子树内的情形，答案可以通过树形 dp 递推得到。

考虑选择其子树外的点的情形。观察原式可以发现，如果一个点的父亲的最优选择不是该点，则其在子树外的 最优选择必然就是其父亲的最优选择。否则，其在子树外的最优选择必然是其父亲的次优选择。由此，可在 dfs 过程中自上而下推出所有点在其子树外的最优答案。

100 分算法

对于有根树，一个点所选择的对应点仅在其子树内的情形，答案可以通过树形 dp 递推得到。

考虑选择其子树外的点的情形。观察原式可以发现，如果一个点的父亲的最优选择不是该点，则其在子树外的 最优选择必然就是其父亲的最优选择。否则，其在子树外的最优选择必然是其父亲的次优选择。由此，可在 dfs 过程中自上而下推出所有点在其子树外的最优答案。

对每个点子树内外的最优答案取最小值，即可得到每个点的答案。

100 分算法

对于有根树，一个点所选择的对应点仅在其子树内的情形，答案可以通过树形 dp 递推得到。

考虑选择其子树外的点的情形。观察原式可以发现，如果一个点的父亲的最优选择不是该点，则其在子树外的 最优选择必然就是其父亲的最优选择。否则，其在子树外的最优选择必然是其父亲的次优选择。由此，可在 dfs 过程中自上而下推出所有点在其子树外的最优答案。

对每个点子树内外的最优答案取最小值，即可得到每个点的答案。

复杂度：$O(n)$。

trails

随机生成一个 $m+1$ 个数的数列，第一个数为 0, 生成第 i 个数时，在前 $i-1$ 个数中等概率选择一个数 k, 则第 i 个数为 $k+1$。每个数均有一个对应的权值，求数列权值和的期望。

$m \leq 21$

对 $m \le 10$ 的算法

可能的数列共有 $m!$ 种，答案就是所有情况下数列的权值和除以情况数。

对 $m \le 10$ 的算法

可能的数列共有 $m!$ 种，答案就是所有情况下数列的权值和除以情况数。

搜索枚举每种情况并求出其权值和，全部加起来除以 $m!$ 即可。

对 $m \le 10$ 的算法

可能的数列共有 $m!$ 种，答案就是所有情况下数列的权值和除以情况数。

搜索枚举每种情况并求出其权值和，全部加起来除以 $m!$ 即可。

复杂度：$O(m!)$。

可以获得 30 分。

对所有 a_i 均相等的算法

由于 a_i 都相等，则对于任意数列，其权值和均相等

对所有 a_i 均相等的算法

由于 a_i 都相等，则对于任意数列，其权值和均相等

直接输出 $m * a_i$ 即可

对所有 a_i 均相等的算法

由于 a_i 都相等，则对于任意数列，其权值和均相等

直接输出 $m * a_i$ 即可

综上，可以获得 50 分

对所有 a_i 均相等的算法

由于 a_i 都相等，则对于任意数列，其权值和均相等

直接输出 $m * a_i$ 即可

综上，可以获得 50 分

100 分算法

一个数列的权值和与数列中数的顺序无关，只与每种数的个数有关

100 分算法

一个数列的权值和与数列中数的顺序无关，只与每种数的个数有关

考虑 dp，状态中需要记录每种数的个数

100 分算法

一个数列的权值和与数列中数的顺序无关，只与每种数的个数有关

考虑 dp，状态中需要记录每种数的个数

可以发现，将得到的数列排序后，相邻两数的差只能为 0 或 1

100 分算法

一个数列的权值和与数列中数的顺序无关，只与每种数的个数有关

考虑 dp，状态中需要记录每种数的个数

可以发现，将得到的数列排序后，相邻两数的差只能为 0 或 1

用二进制序列维护原数列的差分数组，即可表示数列中每种数的个数

100 分算法

一个数列的权值和与数列中数的顺序无关，只与每种数的个数有关

考虑 dp，状态中需要记录每种数的个数

可以发现，将得到的数列排序后，相邻两数的差只能为 0 或 1

用二进制序列维护原数列的差分数组，即可表示数列中每种数的个数

100 分算法

$f(i,S)$ 表示生成了 i 个数，已有数列的差分为 S 的方案数

则每种数列出现的概率 $P_S = \frac{f(m,S)}{m!}$

100 分算法

$f(i,S)$ 表示生成了 i 个数，已有数列的差分为 S 的方案数

则每种数列出现的概率 $P_S = \frac{f(m,S)}{m!}$

转移时枚举前一个数的大小，修改状态即可

100 分算法

$f(i, S)$ 表示生成了 i 个数，已有数列的差分为 S 的方案数

则每种数列出现的概率 $P_S = \frac{f(m,S)}{m!}$

转移时枚举前一个数的大小，修改状态即可

事先预处理出 tot_S 表示状态 S 所表示的数列的权值和，最后的答案即为

$$\sum_S P_S tot_S$$

复杂度：$O(m \cdot 2^m)$。注意做除法时用乘法逆元计算。