康托展开初步

高义雄

July. 5th, 2021

变进制数

$\underline{1}$ $\underline{(a_i)}$

- 给定无穷数集 $S=\{a_0,a_1,a_2,\cdots,a_n,\cdots\}$, $\underline{a_0=1}$
- 定义变进制数： 第 i 位的**单位**是上一位**单位**的 a_i 倍
- 要求变进制数 $(A)_s=\overline{x_nx_{n-1}\cdots x_1x_0}$, $0\le x_i<\underline{a_{i+1}}$
- k 进制数： $\forall i>0,\ a_i=k$

变进制数

- 直观的形式：$(A)_s = \overline{x_n x_{n-1} \cdots x_1 x_0}\,, 0 \le x_i < a_{i+1}$

$$A = \sum_{i=0}^{n} \left(x_i \prod_{j=0}^{i} a_j \right)$$

$$= x_n \prod_{i=0}^{n} a_i + x_{n-1} \prod_{i=0}^{n-1} a_i + \cdots + x_1 a_1 a_0 + x_0 a_0$$

广义康托展开

k → ∞.

(ans + x[i]) × k

$\begin{cases} A \% a_1 \\ A / = a_1 \end{cases}$

- 变进制数下的进制变换
- 对于两个变进制 S_1, S_2 ，给出 $(A)_{S_1}$ ，求 $(A)_{S_2}$
- 一种可操作的思路是 $(A)_{S_1} - (A)_{10} - (A)_{S_2}$
- 考虑一下复杂度和实现方式

广义康托展开

- 变进制 $S = \{a_0, a_1, a_2, \cdots, a_n, \cdots\}, a_0 = 1$
- 记号 $(A)_s = \overline{x_n x_{n-1} \cdots x_1 x_0}, (A)_{10} = A$

```
//(A)s -> (A)10
for (int i = n; ~i; --i) A = (A + x[i]) * a[i];
//(A)10 -> (A)s
for (int i = 0; A; ++i)
    x[i] = A % a[i + 1], A = A / a[i + 1];
```

康托展开

- 康托展开：将任意**十进制自然数** $(\boldsymbol{A})_{\mathbf{10}}$ 变换为**阶乘进制数** $(\boldsymbol{A})_{!}$
- 阶乘进制数：要求 $a_i = i$
- 直观的形式：若 $(A)_! = \overline{x_n x_{n-1} \cdots x_1 x_0}$，有

$$A = x_n * n! + x_{n-1} * (n-1)! + \cdots + x_2 * 2! + x_1 * 1! + x_0 * 1!$$

- 要求 $\forall i \in [0, n], 0 \le x_i < i + 1$，此时**康托展开唯一**

逆康托展开

- 将任意**阶乘进制数** $(A)_!$ 变换为**十进制自然数** $(\boldsymbol{A})_{\mathbf{10}}$

- 实现思路与广义康托展开一致，注意这里 $\prod_{i=0}^{n} a_i = i!$

```
//Cantor Expansion
for (int i = 0; A; ++i)
    x[i] = A % a[i - 1], A = A / a[i - 1];
//Inverse Contor Expansion
for (int i = n; ~i; --i) A = (A + x[i]) * i;
```

(i+1)　(i+1)

引例 - 火星人

- 给定 $1, \cdots, n$ $(n \le 10^3)$ 的排列 $p_1, p_2, \cdots, p_n$ 和正整数 k $(k \le 100)$
- 计算当前排列的字典序排名 $rank$
- 输出 $1, \cdots, n$ 对应的排名为 $rank + k$ 的排列，保证答案存在
- **Bonus：** $n \le 10^5, k \le 10^{18}$

排列 → 变 → 十

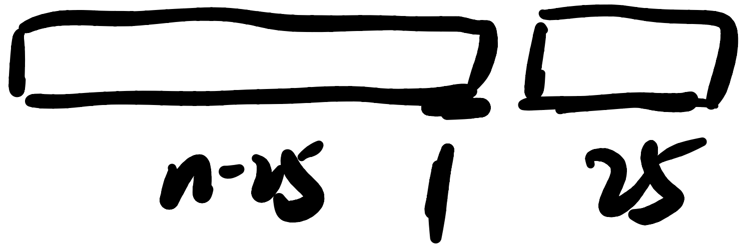
排列与变进制数

- 任何一个 $1, \cdots, n$ 的排列 $p_1, p_2, \cdots, p_n$ 都对应唯一的阶乘进制数
- 构造双射变换 $f: P \to S_!:\ x_i = rank\_suf(p_i)$
- 即 p_i 在后缀 $\{p_i, p_{i+1}, \cdots, p_n\}$ 中的排名，从 0 开始
- 思考一下构造的正确性

实现

- 排列 ⇒ 阶乘进制数：维护数集，支持插入/删除，查询排名
- 阶乘进制数 ⇒ 排列：维护数集，支持查询排名为 k 的数并删除
- 暴力都是 $O(n^2)$，树状数组 $O(n\log^2 n)$，线段树 $O(n\log n)$
- 代码实现见板书

板板



8·25

5×10^4

- 给定 $1,\cdots,n$ $(n \le 10^5)$ 的排列 $p_1,p_2,\cdots,p_n$ 和一个正整数α：

- 操作 1：将当前排列排名增加 k $(k \le 10^{18})$
- 操作 2：输出当前排列对应的 $\sum_{i=1}^{n} p_i \times \alpha^i$ mod 998244353

- Tips：$20! \approx 2 \times 10^{18}$, $25! \approx 1.5 \times 10^{25}$

- **Bonus：** 操作 1 改为继承历史版本，操作 2 中 ~~α 随询问给出~~。

```cpp
#include <bits/stdc++.h>
#define N 100007
#define ls (rt << 1)
#define rs (rt << 1 | 1)
#define mid ((l + r) >> 1)
using namespace std;
typedef long long ll;

int c[N << 2];

inline void pushup(int rt) {
  c[rt] = c[ls] + c[rs];
}

void add(int rt, int l, int r, int p, int x) {
  if (l == r) {c[rt] += x; return;}
  if (p <= mid) add(ls, l, mid, p, x);
  else add(rs, mid + 1, r, p, x);
  pushup(rt);
}

int sum(int rt, int l, int r, int L, int R) {
  if (L <= l && r <= R) return c[rt];
  int res = 0;
  if (L <= mid) res += sum(ls, l, mid, L, R);
  if (R > mid) res += sum(rs, mid + 1, r, L, R);
  return res;
}

int query(int rt, int l, int r, int cnt) {
  if (l == r) return l;
```

```cpp
    if (cnt <= c[ls]) return query(ls, l, mid, cnt);
    return query(rs, mid + 1, r, cnt - c[ls]);
}

ll n, dlt, a[N], x[N];

int main() {
    scanf("%lld%lld", &n, &dlt);
    for (int i = n; i; --i) scanf("%lld", &a[i]);
    //Cantor Expansion
    for (int i = 1; i <= n; ++i) {
        x[i] = sum(1, 1, n, 1, a[i]);
        add(1, 1, n, a[i], 1);
    }
    //Add Rank
    x[1] += dlt;
    for (int i = 1; i <= n; ++i) {
        if (x[i] < i) break;
        x[i + 1] += x[i] / i;
        x[i] = x[i] % i;
    }
    //Inverse Cantor Expansion
    for (int i = n; i; --i) {
        a[i] = query(1, 1, n, x[i] + 1);
        add(1, 1, n, a[i], -1);
    }
    for (int i = n; i; --i) printf("%lld ", a[i]);
    return 0;
}
```

逆序对

- 给定 $1,\cdots,n$ $(n \le 10^5)$ 的排列 $p_1,p_2,\cdots,p_n$ ：
- 操作 1：将当前排列排名增加 k $(k \le 10^{18})$
- 操作 2：输出当前排列的逆序数 mod 998244353
- Tips：$20! \approx 2 \times 10^{18}$, $25! \approx 1.5 \times 10^{25}$

$n \le 1e5$

多重集排列

- 给定多重集 $S = \{a_1, \cdots, a_n\}$ 的排列 $p_1, p_2, \cdots, p_n$
- 求该多重集有多少个本质不同的排列，字典顺序小于当前排列
- 答案对某给定常数 m 取膜，不保证质数

CRT

$\sum_k \sum_{a<p_k} P \cdot \frac{n_k}{N} = \sum_k \frac{P}{N} \sum_{a_i<p_k} n_i$

$N = n - k$

多重集排列数

$P = \frac{N!}{n_1! n_2! \cdots n_n!}$

$P' = P \cdot \frac{N+1}{n_k+1}$

$a_k \quad n_k - 1$

$\frac{\frac{(N-1)!}{n_1! \cdots (n_k-1)! \cdots}}{\frac{N!}{n_1! \cdots n_n!}} = \frac{n_k}{N}$

$\sqrt{[POI]}$ PER.

谢谢大家

More on blog.gyx.me