「杂题记录」「CTSC2017」吉夫特

Jiayi Su (ShuYuMo)

2021-01-20 15:35:06

给出一个长度为 n 的数列 A_i ，求有多少个长度 k 的子序列 A' $(k \geq 2)$ 满足：

$$\prod_{i=1}^{k-1} \binom{A'_i}{A'_{i+1}} > 0 \pmod 2$$

$n \leq 211985, A_i \leq 233333$。原题保证 A_i 互不相同，但是不重要。

分析

根据 Lucas 定理，就是求有多少 A 的子序列 A' 满足:

$$\forall i \in [1, k-1]\ S(A_i) \subseteq S(A_{i+1})$$

$S(x)$ 表示二进制数 x 表示的集合。

这东西可以直接 dp ，设 $f(i)$ 表示以 i 结尾的合法子序列有多少：

$$f(n) \sum_{i=1}^{n-1} f(i)[A_i \operatorname{and} A_n = A_n]$$

直接暴力枚举是 $\mathcal{O}(n^2)$ 的。

考虑类似于分块一样的优化，考虑将 A_i 拆开，设 A_i 二进制下的前 9 位为 x, 后 9 位为 y。$g(x, y)$ 表示前九位恰好为 x ，后九位是 y 的子集的 A_i 的对应 $f(i)$ 之和。

考虑维护这个东西，求出一个 $g(i)$ 后枚举子集更新。

考虑使用这个东西，在求一个 $g(i)$ 时，枚举子集求出。

成功均摊了复杂度。总复杂度 $\mathcal{O}(2^9 n)$

```
const int _ = 241985;
const int MOD = 1e9 + 7;
int A[_], n, f[_];
int g[1 << 10][1 << 10]; // g[x][y]: 当前，所有满足 A_i 的前 9 位为 x ，后 9 位为 y 的超集。
inline int & reduce(int &x) { if(x >= MOD) x-= MOD; if(x < 0) x += MOD; return x; }
int main(){
    ios::sync_with_stdio(false);
    cin >> n; rep(i, 1, n) cin >> A[i]; // 要求前面的数字为后面的超集。
    register int LB = (1 << 9) - 1;
    register int All = ((1 << 9) - 1);
    f[1] = 1;
    g[A[1] >> 9][0] += 1;
```

```
    register int S0 = A[1] & LB; for(register int S = (S0); S; S = (S - 1) & (S0)) reduce(g[A[1] >> 9][
    register int $1;
    rep(i, 2, n) {
        int now = ((1 << 9) - 1) ^ (A[i] >> 9);
        int &ans = f[i] = 1 ;
        $1 = A[i] >> 9;
        reduce(ans += g[$1][A[i] & LB]);
        for(int S = now; S; S = (S - 1) & (now)) reduce(ans += g[S | $1][A[i] & LB]);
        reduce(g[$1][0] += ans);
        for(int S = (A[i] & LB); S; S = (S - 1) & (A[i] & LB)) reduce(g[$1][S] += ans);
    }
    int Ans = 0;
    for(int i = 1; i <= n; i++) reduce(Ans += f[i]); reduce(Ans += MOD - n);
    cout << Ans << endl;
    return 0;
}
```