

memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memseto**
Proudly powered by **Typecho**

洛谷5162 WD与积木

2019-01-27 | 题解

用 f_i 表示 i 个时的高度之和, g_i 表示 i 个时的方案数, 显然:

$$\begin{aligned} g_n &= \sum_{i=1}^n \binom{n}{i} g_{n-i} \\ f_n &= g_n + \sum_{i=1}^n \binom{n}{i} f_{n-i} \\ ans_n &= f_n \times g_n^{-1} \end{aligned}$$

其实到这里可以直接分治 NTT, 不过我们考虑多项式求逆的做法。首先把 $\binom{n}{i}$ 拆开, 然后设 $F_n = f_n/n!$, $G_n = g_n/n!$, $H_n = 1/n!$ 则

$$\begin{aligned} G &= (2 - H)^{-1} \\ F &= (G - 1)(2 - H)^{-1} \\ ans_n &= F_n G_n^{-1} \end{aligned}$$

(推导时需要注意常数项系数, $f_0 = 0$, $g_0 = h_0 = 1$ 。)

代码:



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)

```
// =====
//  author: memset0
//  date: 2019.01.27 22:40:38
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 4e5 + 10, mod = 998244353;
int T, n;
int a[N], f[N], g[N], h[N], w[N], rev[N];

int inv(int x) { return !x || x == 1 ? 1 : (ll)(mod - mod / x) * inv(mod % x)
inline int fpow(int a, int b) { int s = 1; for (; b >= 1, a = (ll)a * a %

inline void ntt(int *a, int lim) {
    for (int i = 0; i < lim; i++) if (i < rev[i]) std::swap(a[i], a[rev[i]]);
    for (int len = 1; len < lim; len <= 1)
        for (int i = 0; i < lim; i += (len << 1))
            for (int j = 0; j < len; j++) {
                int x = a[i + j], y = (ll)w[j + len] * a[i + j + len] % mod;
                a[i + j] = (x + y) % mod, a[i + j + len] = (x - y + mod) % mo
            }
}
```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memseto](#)
Proudly powered by [Typecho](#)

```

}

inline int init(int len) {
    int lim = 1, k = 0;
    while (lim < len) lim <= 1, ++k;
    for (int i = 0; i < lim; i++) rev[i] = (rev[i >> 1] >> 1) | ((i & 1) << (
    for (int len = 1, wn; len < lim; len <= 1) {
        wn = fpow(3, (mod - 1) / (len << 1)), w[len] = 1;
        for (int i = 1; i < len; i++) w[i + len] = (ll)w[i + len - 1] * wn %
    }
    return lim;
}

void polyMul(int *f, int *g, int len) {
    static int a[N], b[N];
    int lim = init(len << 1), inv_lim = inv(lim);
    for (int i = 0; i < lim; i++) a[i] = f[i], b[i] = g[i];
    ntt(a, lim), ntt(b, lim);
    for (int i = 0; i < lim; i++) a[i] = (ll)a[i] * b[i] % mod;
    std::reverse(a + 1, a + lim), ntt(a, lim);
    for (int i = 0; i < lim; i++) f[i] = (ll)a[i] * inv_lim % mod;
}

void polyInv(int *f, int *g, int n) {
    static int a[N], b[N];
    g[0] = inv(f[0]);
    for (int len = 2; (len >> 1) < n; len <= 1) {
        int lim = init(len << 1), inv_lim = inv(lim);
        for (int i = 0; i < len; i++) a[i] = f[i], b[i] = g[i];
        ntt(a, lim), ntt(b, lim);
        for (int i = 0; i < lim; i++) a[i] = (ll)a[i] * b[i] % mod * b[i] % m
        std::reverse(a + 1, a + lim), ntt(a, lim);
        for (int i = 0; i < len; i++) g[i] = ((g[i] << 1) - (ll)a[i] * inv_li
    }
}

void main() {

```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memset0
Proudly powered by Typecho

```
read(T);
for (int i = 1; i <= T; i++) read(a[i]), n = std::max(n, a[i] + 1);
h[0] = h[1] = 1;
for (int i = 2; i < n; i++) h[i] = (1ll)(mod - mod / i) * h[mod % i] % mod;
for (int i = 2; i < n; i++) h[i] = (1ll)h[i] * h[i - 1] % mod;
for (int i = 1; i < n; i++) if (h[i]) h[i] = mod - h[i];
polyInv(h, g, n);
for (int i = 1; i < n; i++) f[i] = g[i];
polyMul(f, g, n);
for (int i = 1; i <= T; i++) print((1ll)f[a[i]] * inv(g[a[i]]) % mod, '\n')
}

} signed main() { return ringo::main(), 0; }
```

[生成函数](#)[分治 NTT](#)[多项式求逆](#)

可以在这里写评论哦 ~



提交评论

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

一句话题解 2019年1月
上一篇 «

洛谷4102 [HEOI2014]林中路径
» 下一篇

© 2017 - 2019 memset0 的博客.

浙ICP备19006255号-1

97702 visits · 24756 visitors · 74.48 W words

关于我
友情链接
文章聚合

在这里输入关键字哦 ~ (回车搜索)

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

