Analysis

记

$$
\begin{aligned}
S &= \sum x \\
A &= \frac{(S^2 - \sum x^2)}{2} \\
cnt_w &= \sum_{i=1}^{n} [p_i = w]
\end{aligned}
$$

则要求

$$
\begin{aligned}
&\left[\prod x_i^2\right] \prod_{i=1}^{n} (A - x_{p_i}(S - x_{p_i})) \\
=&\left[\prod x_i^2\right] \prod_{i=1}^{n} (A - x_{p_i} S + x_{p_i}^2) \\
=&\left[\prod x_i^2\right] A^{cnt_0} \prod_{w=1}^{n} (A - x_w S + x_w^2)^{cnt_w}
\end{aligned}
$$

显然，x_w 的次数至多为 2。于是考虑展开 $(A - x_w S + x_w^2)^{cnt_w}$，展开出来 x_w 的次数 > 2 的项全部舍弃：（为方便，把 cnt_w 记作 c）

$$
\begin{aligned}
&(A - x_w S + x_w^2)^c \\
=&A^c \text{(あ)} - \binom{c}{1} x_w S A^{c-1} \text{(い)} + \binom{c}{2} x_w^2 S^2 A^{c-2} \text{(う)} + \binom{c}{1} x_w^2 A^{c-1} \text{(え)}
\end{aligned}
$$

通过 dp 来计算将 $\left[\prod x_i^2\right] \prod_{w=1}^{n} (A - x_w S + x_w^2)^{cnt_w}$ 展开后的各项系数。
设最终选（い）的有 j 个，选（う）的有 k 个，选（え）的有 l 个，则选（あ）的有 $n-j-k-l$ 个。设这一项的系数为 $dp_{j,k,l}$。
考虑最终的答案为

$$
\left[\prod x_i^2\right] \sum_{j=0}^{n} \sum_{k=0}^{n-j} \sum_{l=0}^{n-j-k} dp_{j,k,l} \prod_{w \in \text{い}} x_w \prod_{w \in \text{う} \cup \text{え}} x_w^2 A^{n-j-l-2k} S^{j+2k}
$$

将 $A^{n-j-l-2k}$ 写成 $\left(\frac{(S^2 - \sum x^2)}{2}\right)^{n-j-l-2k}$ 并用二项式定理展开得

$$
\begin{aligned}
&\frac{1}{2^{n-j-l-2k}} dp_{j,k,l} \prod_{w \in \text{い}} x_w \prod_{w \in \text{う} \cup \text{え}} x_w^2 \left(S^2 - \sum x^2\right)^{n-j-l-2k} S^{j+2k} \\
=&\frac{1}{2^{n-j-l-2k}} \sum_{u=0}^{n-j-l-2k} (-1)^{n-j-l-2k-u} \binom{n-j-l-2k}{u} dp_{j,k,l} \prod_{w \in \text{い}} x_w \prod_{w \in \text{う} \cup \text{え}} x_w^2 \left(\sum x^2\right)^{n-j-l-2k-u} S^{2u+j+2k}
\end{aligned}
$$

考虑如何计算。
使用组合意义，将 $\prod_{w \in \text{い}} x_w \prod_{w \in \text{う} \cup \text{え}} x_w^2$ 这些全部填完后，有 j 个 x 次数为 1，有 v 个 x 次数为 2，有 $n-j-k-l$ 个 x 次数为 0。
显然先将 $(\sum x^2)^{n-j-l-2k-u}$ 全部填到次数为 0 的 x 上（因为它们填且仅能填到次数为 0 的 x 的地方），有

$$
\binom{n-j-k-l}{n-j-l-2k-u} (n-j-l-2k-u)!
$$

种方案。
此时还剩下 $n-j-k-l-(n-j-l-2k-u) = k+u$ 个 x 次数为 0。考虑现在有 $S^{2u+j+2k}$ 用来填满剩下的空余（$k+u$ 个次数为 0 的 x，每个有 2 个空位；j 个次数为 1 的 x，每个有 1 个空位）。
不妨将所有 $2u+2k+j$ 个空位先拿出来排成一列，把颜色（即 x 的下标）相同的空位看做不同的，用 $2u+2k+j$ 个 S 两两匹配并填入。因为每个空位的颜色都是确定的，所以方案数实际上只与每个空位与哪个 S 匹配有关，也就是 $(2u+2k+j)!$ 种方案。

然后每个次数为 0 的位置两个空位顺序无关，于是要除以 2^{u+k}。
所以最终答案即为

$$\sum_{j=0}^{n}\sum_{k=0}^{n-j}\sum_{l=0}^{n-j-k}\sum_{u=0}^{n-j-l-2k} dp_{j,k,l}(-1)^{n-j-l-u}\binom{n-j-l-2k}{u}\binom{n-j-k-l}{n-j-l-2k-u}(n-j-l-2k-u)!\frac{(2u+2k+j)!}{2^{u+k+n-j-l-2k}}$$

考虑记 $q=j+l$，在 dp 状态中合并 j,l 为 q。因为 (い) 和 (え) 两种转移的系数相同，都为 $\binom{c}{1}=c$，所以可以合并。可以理解为把 $-\binom{c}{1}x_wSA^{c-1}$(い) $+\binom{c}{1}x_w^2A^{c-1}$(え) 合并成 $\binom{c}{1}x_w(x_w-S)A^{c-1}$，在最后计算答案时再枚举 (い) 的集合（实际上就是枚举 j 并使用二项式定理展开，也就是乘以 $(-1)^j\binom{q}{j}$），则式子变为 $\sum_{q=0}^{n}\sum_{k=0}^{n-q}\sum_{u=0}^{n-q-2k}\sum_{j=0}^{q} dp_{q,k}(-1)^{n-q-u+j}\binom{q}{j}\binom{n-j-l-2k}{u}\binom{n-q-k}{n-q-2k-u}(n-q-2k-u)!\frac{(2u+2k+j)!}{2^{u+n-q-k}}$ 发现与 j 有关的系数只与 q 和 $k+u$ 的值有关。记 $k+u=\text{く}$，将确定 $\text{く},q$ 时的与 j 有关的系数记作 $coe_{\text{く},q}=\sum_{j=0}^{q}(-1)^{-q+j}\binom{q}{j}$。
先 $O(n^3)$ 预处理出所有的 coe，再 $O(n^3)$ 枚举 q,k,u 的值并计算答案，这时不需要再枚举 j 了，只需要乘以 $coe_{k+u,q}$ 即可。
总时间复杂度 $O(n^3)$。

Code

```cpp
#include<bits/stdc++.h>
using namespace std;

namespace MyNamespace {
typedef long long ll;

inline namespace MyIO {
    inline ll rd() {
        char ch = getchar();
        ll s = 0, w = 1;
        while (!isdigit(ch)) {
            if (ch == '-') w = -1;
            ch = getchar();
        }
        while (isdigit(ch)) {
            s = (s << 3) + (s << 1) + (ch ^ 48);
            ch = getchar();
        }
        return (s * w);
    }
    template<typename T>
    inline void wr(T x) {
        if (x < 0) x = -x, putchar('-');
        if (x > 9) wr(x / 10);
        putchar(x % 10 + 48);
    }
    inline void wrsp() {
        // pass
    }
    template<typename T, typename... Args>
    inline void wrsp(T x, Args... args) {
        wr(x), putchar(' '), wrsp(args...);
    }
    inline void wrln() {
        putchar('\n');
    }
    template<typename T>
```

```cpp
    inline void wrln(T x) {
        wr(x), putchar('\n');
    }
    template<typename T, typename... Args>
    inline void wrln(T x, Args... args) {
        wr(x), putchar(' '), wrln(args...);
    }
} // namespace MyIO

inline namespace Base {
    #define bug(x) << #x << '=' << (x) << ' '
    #define siz(A) ll((A).size())

    template<typename T>
    constexpr inline void Max(T &x, T y) {
        x = max(x, y);
    }
    template<typename T>
    constexpr inline void Min(T &x, T y) {
        x = min(x, y);
    }
} // namespace Base

constexpr int fn = 1500;
constexpr int N = fn + 10;

inline namespace ModInt {
    constexpr ll MO = 998244353;
    constexpr inline ll moa(ll x) {
        return (x >= MO ? x - MO : x);
    }
    constexpr inline ll moi(ll x) {
        return (x < 0 ? x + MO : x);
    }
    constexpr inline ll mo(ll x) {
        return moi(x % MO);
    }
    constexpr inline ll& Moa(ll &x) {
        return x = moa(x);
    }
    constexpr inline ll& Moi(ll &x) {
        return x = moi(x);
    }
    constexpr inline ll& Mo(ll &x) {
        return x = mo(x);
    }
    constexpr inline ll qpow(ll x, ll y) {
        Mo(x);
        ll k = 1;
        while (y) {
            if (y & 1) (k *= x) %= MO;
            (x *= x) %= MO;
            y >>= 1;
        }
        return k;
    }
```

```cpp
    constexpr inline ll get_inv(ll x) {
        return qpow(x, MO - 2);
    }
    constexpr ll MOI2 = get_inv(2);
    inline string to_frac(ll k) {
        for (ll v = 1; v <= 1000; ++v) {
            ll u = k * v % MO;
            if (u <= 1000) return to_string(u) + '/' + to_string(v);
            else if (MO - u <= 1000) return to_string(u - MO) + '/' + to_string(v);
        }
        return "NaN";
    }
} // namespace ModInt

inline namespace ModIntBase {
    ll inv[N], jc[N], ji[N], pw2I[N];
    void init_inv() {
        jc[0] = 1;
        for (int i = 1; i <= fn; ++i)
            jc[i] = jc[i - 1] * i % MO;

        ji[fn] = get_inv(jc[fn]);
        for (int i = fn; i >= 1; --i)
            ji[i - 1] = ji[i] * i % MO;

        inv[0] = 1;
        for (int i = 1; i <= fn; ++i)
            inv[i] = jc[i - 1] * ji[i] % MO;

        pw2I[0] = 1;
        for (int i = 1; i <= fn; ++i)
            pw2I[i] = pw2I[i - 1] * MOI2 % MO;
    }
    inline ll binom(int n, int m) {
        return jc[n] * ji[n - m] % MO * ji[m] % MO;
    }
} // namespace ModIntBase

int n, a[N], b[N];

ll dp[2][N][N], (*f)[N], (*g)[N];
ll coe[N][N];

/*
字母不够用了怎么办 /ll
好想用平假名作为变量名啊 (
算了，用平假名命名情况吧
别问我为啥不用甲乙丙丁
q = j + 1
k
*/

void NamespaceMain() {
    init_inv();

    n = rd();
```

```cpp
    for (int i = 1; i <= n; ++i)
        a[i] = rd(), ++b[a[i]];

    f = dp[0], g = dp[1];
    f[0][0] = 1;
    for (int i = 1; i <= n; ++i) {
        for (int q = 0; q <= i; ++q)
            for (int k = 0; k <= i - q; ++k)
                g[q][k] = 0;
        for (int q = 0; q <= i - 1; ++q)
            for (int k = 0; k <= i - 1 - q; ++k) {
                if (!f[q][k]) continue;

                // (あ)
                Moa(g[q][k] += f[q][k]);

                // (い) && (え)
                if (b[i] >= 1)
                    (g[q + 1][k] += f[q][k] * b[i]) %= MO;

                // (う)
                if (b[i] >= 2)
                    (g[q][k + 1] += f[q][k] * ll(b[i] * (b[i] - 1) / 2)) %= MO;
            }
        swap(f, g);
    }

    for (int ku = 0; ku <= n; ++ku) // k + u
    for (int q = 0; q <= n; ++q)
        for (int j = 0; j <= q; ++j)  {
            ll val = binom(q, j) * jc[2 * ku + j] % MO;
            if ((-q + j) & 1)
                Moi(coe[ku][q] -= val);
            else
                Moa(coe[ku][q] += val);
        }

    ll ans = 0;
    for (int q = 0; q <= n; ++q)
    for (int k = 0; k <= n - q; ++k)
    for (int u = 0; u <= n - q - k * 2; ++u) {
        ll val = f[q][k] * binom(n - q - 2 * k, u) % MO
        * jc[n - q - k] % MO * ji[k + u] % MO
        * pw2I[u + n - q - k] % MO
        * coe[k + u][q] % MO;
        // cerr bug(q) bug(j) bug(l) bug(k) bug(u) bug(f[q][k]) bug(val) bug(to_frac(val)) bug((n - q - u + j) & 1) bug(ans) << endl;
        if ((n - u) & 1)
            Moi(ans -= val);
        else
            Moa(ans += val);
    }

    wr(ans);
}
}
```

```
int main() {
    MyNamespace::NamespaceMain();
    return 0;
}
```