

E6H 集合和集合的和还是集合

题目描述

集合计算 $A + B = \{x + y | x \in A, y \in B\}$

给定整数集合 S ，求 $S + S$ 有多少个不同的整数

解题分析

可以联想到 $a^x \times a^y = a^{x+y}$

那么，对于两组不同的 $x_1 + y_1 = k, x_2 + y_2 = k$ ，若 x_1, x_2, y_1, y_2 均为 S 中的元素，那么它们在 $S + S$ 中算作一个数，把它们看作多项式的项，

$$a^{x_1} \times a^{y_1} + a^{x_2} \times a^{y_2} = 2a^k$$

故，我们可以将 S 的项 x_i 看作 a^{x_i} ，那么 $S \otimes S$ 中非零项的个数即为 $S + S$ 中不同整数的个数

当然，注意到数据范围 $-5 \times 10^4 \leq x_i \leq 5 \times 10^4$ ，故我们可以将所有的 x_i 加上 5×10^4 后作为多项式项的幂进行计算

代码示例

```
#include <bits/stdc++.h>

using namespace std;

using ll = long long;
using db = long double; // or double, if TL is tight
using str = string;
using u32 = unsigned int;
using u64 = unsigned long long;
using u128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<db, db>;
#define fi first
#define se second

#define vc vector
using vi = vc<int>;
using vb = vc<bool>;
using vll = vc<ll>;
using vd = vc<db>;
```

```

using vs = vc<str>;
using vch = vc<char>;
using vpi = vc<pi>;
using vpl = vc<pl>;
using vpdd = vc<pdd>;
using vvi = vc<vc<int>>;
using vvll = vc<vc<ll>>;
using vvch = vc<vc<char>>;
using vvb = vc<vc<bool>>;
using vvp = vc<vc<pi>>;
using vvppl = vc<vc<pl>>;
using vvvi = vc<vc<vc<int>>>;

const int mod = 998244353; // 1e9 + 7;
const int INF = 0x3f3f3f3f;
const ll BIG = 1e18; // not too close to LLONG_MAX
const db PI = acos((db)-1);
const int dx[4]{1, 0, -1, 0}, dy[4]{0, 1, 0, -1}; // for
every grid problem!!
mt19937
rng32((uint32_t)chrono::steady_clock::now().time_since_epo
ch().count());
mt19937_64
rng64((uint64_t)chrono::steady_clock::now().time_since_epo
ch().count());

#define cl(x, y) memset(x, y, sizeof(x))
#define de(x) cerr << #x << " = " << x << " "
#define del(x) cerr << #x << " = " << x << endl

#define overload3(a, b, c, d, ...) d
#define rep1(a) for(int i = 0; i < (a); i++)
#define rep2(i, a) for (int i = 0; i < (a); i++)
#define rep3(i, a, b) for (int i = (a); i <= (b); i++)
#define per1(a) for(int i = (a) - 1; i >= 0; i--)
#define per2(i, a) for (int i = (a) - 1; i >= 0; i--)
#define per3(i, a, b) for (int i = (a); i >= (b); i--)
#define rep(args...) overload3(args, rep3, rep2, rep1)
(args)
#define per(args...) overload3(args, per3, per2, per1)
(args)
#define for_subset(t, s) \
    for (ll t = (s); t >= 0; t = (t == 0 ? -1 : (t - 1) &
(s)))

#define co cout
#define dl '\n'
#define ret return
#define pb push_back
#define tcT template <class T
#define YES cout << "YES" << endl
#define Yes cout << "Yes" << endl

```

```

#define NO cout << "NO" << endl
#define No cout << "No" << endl
#define INT(args...) \
    int args; \
    rd(args)
#define LL(args...) \
    ll args; \
    rd(args)

#define all(s) s.begin(), s.end()
#define rall(s) s.rbegin(), s.rend()
#define sz(v) int(v.size())
#define SUM(a) accumulate(all(a), 0ll)
#define MIN(v) (*min_element(all(v)))
#define MAX(v) (*max_element(all(v)))
#define unisort(v) \
    sort(all(v)); \
    v.resize(unique(all(v)) - v.begin())

int popcnt(int x) { return __builtin_popcount(x); }
int popcnt(u32 x) { return __builtin_popcount(x); }
int popcnt(ll x) { return __builtin_popcountll(x); }
int popcnt(u64 x) { return __builtin_popcountll(x); }

#define LOWBIT(x) ((x) & (-x))
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 1, 2)
int topbit(int x) { return (x == 0 ? -1 : 31 - __builtin_clz(x)); }
int topbit(u32 x) { return (x == 0 ? -1 : 31 - __builtin_clz(x)); }
int topbit(ll x) { return (x == 0 ? -1 : 63 - __builtin_clzll(x)); }
int topbit(u64 x) { return (x == 0 ? -1 : 63 - __builtin_clzll(x)); }
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 0, 2)
int lowbit(int x) { return (x == 0 ? -1 : __builtin_ctz(x)); }
int lowbit(u32 x) { return (x == 0 ? -1 : __builtin_ctz(x)); }
int lowbit(ll x) { return (x == 0 ? -1 : __builtin_ctzll(x)); }
int lowbit(u64 x) { return (x == 0 ? -1 : __builtin_ctzll(x)); }
tcT > T cdiv(T &a, T &b) {
    return a / b + ((a ^ b) > (T)(0) && a % b);
} // divide a by b rounded up
tcT > T fdiv(T &a, T &b) {
    return a / b - ((a ^ b) < (T)(0) && a % b);
} // divide a by b rounded down

```

```

tcT, class S > inline ostream &operator<<(ostream &i,
const pair<T, S> &a) { return i << a.first << ' ' <<
a.second; }
tcT, class S > inline istream &operator>>(istream &i,
pair<T, S> &a) { return i >> a.first >> a.second; }
tcT > inline ostream &operator<<(ostream &i, const
vector<T> &a) {
    if (a.size())
        i << a[0];
    for (int j = 1; j < int(a.size()); j++)
        i << ' ' << a[j];
    return i;
}
tcT > inline istream &operator>>(istream &i, vector<T> &a)
{
    for (auto &j : a)
        i >> j;
    return i;
}
void err(istream_iterator<string> it) {
it.~istream_iterator(); }
template <typename T, typename... Args>
void err(istream_iterator<string> it, T a, Args... args) {
    cerr << *it << " = " << a << endl;
    err(++it, args...);
}
void wt() { cout << '\n'; }
tcT, class... Args > void wt(const T &a, const Args
&...args) {
    cout << a;
    (void)(cout << ... << (cout << ' ', args));
    cout << '\n';
}
tcT > void wt(vector<vector<T>> &a) {
    for (auto &i : a)
        wt(i);
}
tcT, class S > void wt(vector<pair<T, S>> &a) {
    for (auto &i : a)
        wt(i);
}
template <class... T>
void rd(T &...a) { (cin >> ... >> a); }

#define LOCAL_TEST freopen("out.txt", "w", stdout)
#define IOS
ios::sync_with_stdio(false);cin.tie(0);cout.tie(0)
#define FIX(x) cout << fixed << setprecision(x)

const int maxn = 1e5 + 5;
const int delta = 50000;

```

```

struct Complex {
    db x, y;

    Complex(db _x = 0.0, db _y = 0.0) { x = _x, y = _y; }

    Complex operator-(const Complex &b) const { return
Complex(x - b.x, y - b.y); }
    Complex operator+(const Complex &b) const { return
Complex(x + b.x, y + b.y); }
    Complex operator*(const Complex &b) const { return
Complex(x * b.x - y * b.y, x * b.y + y * b.x); }
};
vector<int> r;
int l = 0;

/**
 * extend the length to power of 2
 * store length in &lim
 * and initialize the array rev
 */
typedef vc<Complex> C;
void INIT(C &a, int n, C &b, int m, int &lim) {
    lim = 1; l = 0;
    while(lim <= n + m) { lim <=< 1; l++; }
    a.resize(lim, Complex());
    b.resize(lim, Complex());
    r.resize(lim);
    rep(i, 0, lim - 1)
        r[i] = (r[i >> 1] >> 1) | ((i & 1) << (l - 1));
}

/**
 * FFT
 * n MUST be power of 2
 * when on == 1, DFT. on == -1, IDFT
 * @param y polynomial
 * @param on DFT/IDFT
 */
void FFT(C &y, int n, int on = 1) {
    rep(i, 0, n - 1)
        if(i < r[i]) swap(y[i], y[r[i]]);
    for(int h = 2; h <= n; h <=< 1) {
        Complex wn(cos(2 * PI / h), sin(on * 2 * PI / h));
        for(int j = 0; j < n; j += h) {
            Complex w(1, 0);
            for(int k = j; k < j + h / 2; k++, w = w * wn)
{
                Complex u = y[k], t = w * y[k + h / 2];
                y[k] = u + t;
                y[k + h / 2] = u - t;
            }
        }
    }
}

```

```

    }
    if(on == -1)
        rep(i, 0, n - 1) y[i].x /= n;
}

inline void INIT() {

}

inline void SOLVE() {

    INT(n);
    vector<Complex> A(maxn);
    rep(n) {
        INT(x);
        A[x + delta].x += 1;
    }
    int lim;
    INIT(A, maxn, A, maxn, lim);
    FFT(A, lim, 1);
    rep(lim) A[i] = A[i] * A[i];
    FFT(A, lim, -1);

    int cnt = 0;
    rep(lim) cnt += (abs(A[i].x) > 1e-5);
    wt(cnt);

}

/*

*/

int main() {

    IOS; FIX(20);

    INIT();
    int tt = 1;
    cin >> tt;
    while(tt--)
        SOLVE();

    // cout << "program ends..." << endl;
    // system("pause");
    return 0;
}

```