**// ========= 筛质数 =========**

bool vis [10000007];

vector<int> primes;

void init(){

// vis[0] = vis[1] = true;

// for (int i = 4; i < maxn; i += 2)

// vis[i] = true;

primes.push\_back(2);

for(int i=3; i<maxn; i+=2){

if(!vis[i])

primes.push\_back(ll(i));

for (int p : primes) {

if (p \* i > maxn-1) break;

vis[p \* i] = true;

if (i % p == 0) break;

}

}

}

**// ========= FFT 傅里叶=========**

#include <algorithm>

#include <cmath>

using namespace std;

const double PI = acos(-1.0);

struct complex {

double r,i;

complex(double \_r = 0.0,double \_i = 0.0)

{r = \_r; i = \_i;}

complex operator +(const complex &b)

{return complex(r+b.r,i+b.i);}

complex operator -(const complex &b)

{return complex(r-b.r,i-b.i);}

complex operator \*(const complex &b)

{return complex(r\*b.r-i\*b.i,r\*b.i+i\*b.r);}

};

void change(complex y[],int len) {

int i,j,k;

for(i = 1, j = len/2;i < len-1; i++)

{

if(i < j)swap(y[i],y[j]);

k = len/2;

while( j >= k) {j -= k;k /= 2;}

if(j < k) j += k;

}

}

void fft(complex y[],int len,int on)

//on==-1 IDFT

{

change(y,len);

for(int h = 2; h <= len; h <<= 1)

{

complex wn(cos(-on\*2\*PI/h),sin(-on\*2\*PI/h));

for(int j = 0;j < len;j+=h)

{

complex w(1,0);

for(int k = j;k < j+h/2;k++)

{

complex u = y[k];

complex t = w\*y[k+h/2];

y[k] = u+t;

y[k+h/2] = u-t;

w = w\*wn;

}

}

}

if(on == -1)

for(int i = 0;i < len;i++) y[i].r /= len;

}

const int MAXN = 200011;

complex x[MAXN \* 4];

LL num[MAXN \* 4];

int a[MAXN];

int main() {

memset(num, 0, sizeof(num));

for (int i = 0;i < N; i++) {

scanf("%d", &a[i]);

num[a[i]]++;

}

sort(a, a+N);

int len\_tmp = a[N-1]+1, len = 1;

while (len < len\_tmp\*2)

len <<= 1;

for (int i=0;i<len;i++)

x[i] = complex(num[i],0);

fft(x, len, 1);//DFT

for(int i = 0;i < len;i++)

x[i] = x[i]\*x[i];

fft(x, len, -1);//IDFT

for (int i = 0;i < len;i++)

num[i] = (LL)round(x[i].r);

//可能要：求组合而不是求排列

num[i] /= 2;

//可能要：扣除a[i]+a[i]的情况

num[a[i]+a[i]]--;

//可能要：扣除带0的特殊情况

Cnt -= (LL)Cnt0 \* (N-1) \* 2LL;// 0+ai=ai && ai+0=ai

printf("%lld\n", Cnt);

return 0;

}