

```

A1 const int MAXN = (int) 1e6 + 5;
const int MOD = (int) 1e9 + 7;

int a[MAXN];
int d[MAXN];
int n;
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    sort(a + 1, a + n + 1);
    if (a[n] <= 0) {
        cout << "1\n";
        return 0;
    }
    int t = n;
    while (t > 2 && a[t-2] + a[t-1] > a[t] && a[t-1] > 0) {
        t--;
    }
    d[t] = (int) 2e9;
    for (int i = 2; i <= n; i++) {
        d[i] = a[i] - a[i-1];
    }
    vector<pair<int, int>> values;
    for (int i = 1; i <= n; i++) {
    }
    sort(all(values));
    set<int> S;
    for (int i = 1; i <= n; i++) {
        S.insert(i);
    }
    int ans = 0;

    int ptr = 0;
    for (int j = t; j <= n; j++) {
        while (ptr < S.size() && values[ptr].first < a[j]) {
            int idx = values[ptr].second;
            S.erase(idx);
            ptr++;
        }
        int i = *prev(S.lower_bound(i));
        ans += (j - i);
    }
    cout << ans << '\n';
    return 0;
}

```

```

B
const int MAXN = (int) 1e6 + 5;
//vector <vector<int>> where [MAXN];
vector <int> where [MAXN];
int a [MAXN];
int n;
int index - lim;
int fenw - max [MAXN];
int pref - max (int p) {
int res = 0;
for (; p > 0; p -= p & -p) {
res = max (res, fenw - max [p]);
} return res;
}
void update - max (int p, int x) {
for (; p <= n; p += p & -p) {
fenw - max [p] = max (fenw - max [p], x);
}
}
void add (int x) {
int prv = 0;
for (int pos : where [x]) {
update - max (prv + 1, pos);
prv = pos;
} index - lim = min (index - lim, prv);
}
int main () {
ios :: sync - with - stdio (0);
cin . tie (0);
cin >> n;
for (int i = 1; i <= n; i++) {
cin >> a [i];
where [a [i]] . pb (i);
}
index - lim = n;
for (int x = 0; x <= n - 1; x++) {
add (x);
int i = 1, ans = 0;
while (i <= n) {
int j;
if (i > index - lim) {
j = n + 1;
} else {
j = pref - max (i);
} ans++;
if (j == i) {
ans -= 1;
break;
} i = j;
}
cout << ans << "\n" [x == n];
}
//cout << (double) clock () / CLOCKS_PER_SEC << endl;
return 0;
}
    
```