

给定一个长度为  $n$  的数组，数组中的每个数分属四个不同种类，每个种类的数取一个，满足下列条件：

$$\max(X_4, X_3) - \min(X_1, X_2) \leq d$$

$$\max(X_1, X_2) - \min(X_4, X_3) \leq d$$

其中  $n \leq 1e6$

```
#include<bits/stdc++.h>
using i8 = signed char;
using u8 = unsigned char;
using i16 = signed short int;
using u16 = unsigned short int;
using i32 = signed int;
using u32 = unsigned int;
using f32 = float;
using i64 = signed long long;
using u64 = unsigned long long;
using f64 = double;
using i128 = __int128_t;
using u128 = __uint128_t;
using f128 = long double;
using namespace std;
const i64 mod = 666623333;
const i64 maxn = 1e5 + 1;
const i64 inf = 0x3f3f3f3f3f3f3f3f;
int w[maxn];
vector<pair<int, int>>k[3];
void solve() {
    int n; std::cin >> n;
    vector<int>w(n + 1);
    for (int i = 1; i <= n; i++)std::cin >> w[i];
    vector<vector<int>>a(5);
    vector<vector<pair<int, int>>>k(3);
    for (int i = 1; i <= 4; i++) {
        a[i].push_back(0);
    }
    for (int i = 1; i <= n; i++) {
        int v; std::cin >> v; a[w[i]].push_back(v);
        k[(w[i] + 1) / 2].push_back({w[i], v});
    }
    for (int i = 1; i <= 4; i++) {
        sort(a[i].begin() + 1, a[i].end());
    }
    int l = 0, r = 1e9;
    auto check = [&](int mid) {
        for (auto [idx, v] : k[2]) {
            int q = ((idx == 3) ? 4 : 3);
            int ind = upper_bound(a[q].begin() + 1, a[q].end(), v) - a[q].begin()
- 1;
            if (ind >= 1) {
                int v2 = a[q][ind]; // v : 34最大值 v2: 34最小值
                int idx1 = lower_bound(a[1].begin() + 1, a[1].end(), v - mid) -
a[1].begin();
                int idx2 = upper_bound(a[2].begin() + 1, a[2].end(), mid + v2) -
a[2].begin() - 1;
```

```

        if (idx1 < a[1].size() and idx2 >= 1 and a[1][idx1] <= a[2]
[idx2])return true;
        idx1 = lower_bound(a[2].begin() + 1, a[2].end(), v - mid) -
a[2].begin();
        idx2 = upper_bound(a[1].begin() + 1, a[1].end(), mid + v2) -
a[1].begin() - 1;
        if (idx1 < a[2].size() and idx2 >= 1 and a[2][idx1] <= a[1]
[idx2])return true;
    }
}
return false;
};
int ans = 0;
while (l <= r) {
    int mid = (l + r) >> 1;
    if (check(mid))r = mid - 1, ans = mid;
    else l = mid + 1;
}
std::cout << ans << "\n";
}
int main() {

    int T; scanf("%d\n", &T);
    while (T--) {
        solve();
    }
    return 0;
}

```