Problem Link:

<https://leetcode.com/problems/rearranging-fruits/?envType=daily-question&envId=2025-08-02>

Solution:

class Solution {

public:

long long minCost(vector<int>& basket1, vector<int>& basket2) {

map<int, int> f;

map<int, int> c1, c2;

int n = basket1.size();

int m = INT\_MAX;

for(int x : basket1)

{

f[x]++;

c1[x]++;

m = min(m, x);

}

for(int x : basket2)

{

f[x]++;

c2[x]++;

m = min(m, x);

}

for(auto p : f)

{

if(p.second % 2 != 0)

return -1;

}

vector<int> v1, e2;

for(auto p : f)

{

int n = p.second / 2;

if(c1[p.first] > n)

{

for(int i = 0; i < c1[p.first] - n; ++i)

v1.push\_back(p.first);

}

else if(c2[p.first] > n)

{

for(int i = 0; i < c2[p.first] - n; ++i)

e2.push\_back(p.first);

}

}

sort(v1.begin(), v1.end());

sort(e2.rbegin(), e2.rend());

long long c = 0;

int s = v1.size();

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

{

c += min((long long)min(v1[i], e2[i]), 2LL \* m);

}

return c;

}

};