

CF468B Two Sets

题目大意

给出 n 个各不相同的数字，将它们分别放入 A 和 B 两个集合中，使它们满足：

- 若数字 x 在集合 A 中，那么数字 $a - x$ 也在集合 A 中；
- 若数字 x 在集合 B 中，那么数字 $b - x$ 也在集合 B 中。

(by **Uranus**)

题解

感觉网上全是并查集的题解。

没有贪心？

感觉贪心比并查集好想啊……

首先我们想到的肯定是开个set大力匹配，然而发现对于一个 x 可能 $a - x$ 和 $b - x$ 都在序列中，于是我们就陷入两难了。

如何解决这个问题呢？

现在我们假设 $a \geq b$ 。

我们每次贪心地选出没有匹配过的数的最小值，设其为 x 。

假设我们发现 $a - x$ 和 $b - x$ 都在序列中且都没有被匹配过。

我们会发现 x 一定与 $a - x$ 匹配。

假设答案是 x 与 $b - x$ 匹配，那也就是说 $a - x$ 不在 A 集合里，所以其在 B 集合里，则与之匹配的是 $b - (a - x) = x + (b - a) \leq x$ ，但由于 x 是序列中的最小数，所以不存在 $b - (a - x)$ 。

代码也很简单：

```
#include <cstdio>
#include <cstring>
#include <set>

using namespace std;

const int maxn = 100005;
```

```

int ans[maxn];

struct EE
{
    int x, id;

    inline bool operator < (const EE& other) const
    {
        return this->x < other.x;
    }
} aa[maxn];

set<EE> ss;

int main()
{
    int n, a, b;
    scanf("%d%d", &n, &a, &b);
    ss.clear();
    bool f = false;
    if(a < b)
    {
        swap(a, b);
        f = true;
    }
    for(int i = 1; i <= n; ++i)
    {
        EE aa;
        scanf("%d", &aa.x);
        aa.id = i;
        ss.insert(aa);
    }
    memset(ans, 0xff, sizeof(ans));
    while(!ss.empty())
    {
        set<EE>::iterator it = ss.begin();
        EE tx = *it;
        tx.x = a - it->x;
        set<EE>::iterator x = ss.lower_bound(tx);
        if(x != ss.end() && x->x + it->x == a)
        {
            ans[x->id] = ans[it->id] = 0;
            if(x->id != it->id)
            {
                ss.erase(x);
                ss.erase(it);
            }
            else
                ss.erase(x);
        }
    }
}

```

```

else
{
    tx.x = b - it->x;
    x = ss.lower_bound(tx);
    if(x != ss.end() && x->x + it->x == b)
    {
        ans[x->id] = ans[it->id] = 1;
        if(x->id != it->id)
            ss.erase(it);
        ss.erase(x);
    }
    else
        return puts("NO"), 0;
}
}
puts("YES");
for(int i = 1; i <= n; ++i)
    printf("%d ", ans[i] ^ f);
return 0;
}

```