#include <cstdio>

#include <algorithm>

using namespace std;

int n, m;

int d[600010];

void add(int p, int s, int e, int a, int b)

{

if (b < a)

return;

if (s == a && e == b)

{

d[p]++;

return;

}

int l = (s+e)>>1;

if (b <= l)

add(p<<1, s, l, a, b);

else

if (a >= l+1)

add((p<<1)|1, l+1, e, a, b);

else

{

add(p<<1, s, l, a, l);

add((p<<1)|1, l+1, e, l+1, b);

}

}

int val(int a)

{

int ret=0;

a += n-1;

while (a)

{

ret += d[a];

a >>= 1;

}

return ret;

}

int a[600010], b[600010];

int main()

{

int p, q, l, v, x, y, i;

scanf("%d%d", &n, &m);

for (i = 1; i <= n; i++)

scanf("%d", a+i);

for (i = 1; i <= m; i++)

scanf("%d", b+i);

sort(a+1, a+n+1);

reverse(a+1, a+n+1);

sort(b+1, b+m+1);

reverse(b+1, b+m+1);

for (i = 1; i < n; i<<=1);

n = i;

for (i = 1; i <= m; i++)

{

if (b[i] > n)

{

printf("NO\n");

return 0;

}

v = val(b[i]);

if (a[b[i]]-v == 0)

{

printf("NO\n");

return 0;

}

if (a[b[i]]-v != a[b[i]+1]-val(b[i]+1))

add(1, 1, n, 1, b[i]);

else

{

p = 1;

q = b[i];

while (q-p > 1)

{

l = (p+q)>>1;

if (a[l]-val(l) == a[b[i]]-v)

q = l;

else

p = l;

}

if (a[p]-val(p) == a[b[i]]-v)

x = p;

else

x = q;

p = b[i];

q = n;

while (q-p > 1)

{

l = (p+q)>>1;

if (a[l]-val(l) == a[b[i]]-v)

p = l;

else

q = l;

}

if (a[q]-val(q) == a[b[i]]-v)

y = q;

else

y = p;

add(1, 1, n, 1, x-1);

add(1, 1, n, y-b[i]+x, y);

}

}

printf("YES\n");

return 0;

}

/\*

10 10

4 4 10 4 7 3 6 5 2 6

5 7 8 3 3 2 4 4 4 10

\*/