

## [hard,math]20180325\_两个集合放均值

NowCoder 牛客网 NOWCODER 头条 今日头条20

给一个包含 $n$ 个整数元素的集合 $a$ ，一个包含 $m$ 个整数元素的集合 $b$ 。

定义magic操作为，从一个集合中取出一个元素，放到另一个集合里，且操作过后每个集合的平均值都大于操作前。

注意以下两点：

- ①不可以把一个集合的元素取空，这样就没有平均值了
- ②值为 $x$ 的元素从集合 $b$ 取出放入集合 $a$ ，但集合 $a$ 中已经有值为 $x$ 的元素，则 $a$ 的平均值不变（因为集合元素不会重复）， $b$ 的平均值可能会改变（因为 $x$ 被取出了）

问最多可以进行多少次magic操作？

输入描述：

- 第一行为两个整数 $n, m$
- 第二行 $n$ 个整数，表示集合 $a$ 中的元素
- 第三行 $m$ 个整数，表示集合 $b$ 中的元素

对于30%的数据，最终结果 $\leq 1$

对于70%的数据，输入中的 $a, b$ 集合元素完全没有重复，即  $|a| + |b| = |a \cup b|$

对于100%的数据， $1 < n, m < 100,000, 0 < a[i], b[i] < 100,000,000$ ，集合 $a$ 中元素互不相同，集合 $b$ 中元素互不相同。

输出描述：

收起答题卡

例1 例2 1 2 3 4 5

保存并

只能从均值大的集合往均值小的集合里放。

取数只能取出现次数等于1的数 放数只能放没出现过的数 从小的数开始放可以使均值小的集合均值上升慢，均值大的集合均值上升快，这样最优。

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
set<int> sa,sb;
ll suma, sumb;
const long double eps = 1e-14;
inline int cmp(long double a, long double b)
{
    if(fabs(a-b) <= eps) return 0;
    return a>b?1:-1;
}
inline long double jz(ll k, int m)
{
    return (long double)k/m;
}
int main()
{
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=0; i<n;++i)
    {
        int t;
        scanf("%d",&t);
        sa.insert(t);
        suma+=t;
    }
    for(int i=0;i<m;++i)
    {
        int t;
        scanf("%d",&t);
        sb.insert(t);
```

```

        sumb+=t;
    }
    if(cmp(jz(suma, n), jz(sumb, m))==-1)
    {
        swap(suma, sumb);
        swap(n, m);
        sa.swap(sb);
    }
    int mx =n;
    int ans = 0;
    for(auto k : sa)
    {
        if(cmp(k, jz(suma, n)) >= 0) break;
        // printf("%d %d\n",n, k);
        if(!sb.count(k)&&cmp(k, jz(sumb, m))>0)
        {
            ++ans;
            sumb+=k;
            suma-=k;
            sb.insert(k);
            --n;++m;
        }
    }
    printf("%d\n", ans);

```