

C1-D题题解

班级：222112
学号：22373100
姓名：郑悦

· 题目

新 · 电子对撞

时间限制: 1000 ms 内存限制: 65536 kb

总通过人数: 85 总提交人数: 96

本题必须使用 C 语言提交。

题目描述

hls 觉得很抽象，为什么做梦也要写算法题。QAQ 于是他一气之下修改了两个电子对撞产生能量的规则。

具体来说：

给定整数序列 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_m ，严格递增的非负整数序列 A_1, A_2, \dots, A_n 和 B_1, B_2, \dots, B_m ，求解如下多项式：

$$\left(\sum_{i=1}^n a_i x^{A_i} \right) + \left(\sum_{i=1}^m b_i x^{B_i} \right)$$

输入格式

第一行一个正整数 t ($1 \leq t \leq 5$)，表示数据组数。

对于每组数据，第一行一个正整数 n ($1 \leq n \leq 10^5$)，含义同题目描述。

第二行 n 个整数 a_1, a_2, \dots, a_n ($0 < |a_i| \leq 10^9$)，含义同题目描述。

第三行 n 个非负整数 A_1, A_2, \dots, A_n ($0 \leq A_i \leq 10^9$)，含义同题目描述。

第四行一个正整数 m ($1 \leq m \leq 10^5$)，含义同题目描述。

第五行 n 个整数 b_1, b_2, \dots, b_m ($0 < |b_i| \leq 10^9$)，含义同题目描述。

第六行 n 个非负整数 B_1, B_2, \dots, B_m ($0 \leq B_i \leq 10^9$)，含义同题目描述。

保证 $A_i > A_{i-1}$ 对 $1 < i \leq n$ 成立， $B_i > B_{i-1}$ 对 $1 < i \leq m$ 成立，
 $A_i = B_j \Rightarrow a_i + b_j \neq 0$ 对 $1 \leq i \leq n, 1 \leq j \leq m$ 成立。

输出格式

对于每组数据，输出三行：

- 第一行一个正整数 k ，表示所得多项式在**合并同类项**后有 k 个系数非 0 项，并设所得多项式为：

$$\sum_{i=1}^k c_i x^{C_i}$$

其中 $C_i > C_{i-1}$ 对 $1 < i \leq k$ 成立。

- 第二行 k 个整数 c_1, c_2, \dots, c_k ，含义同上式。
- 第三行 k 个非负整数 C_1, C_2, \dots, C_k ，含义同上式。

注意点：

多项式合并同类项，因为题目的输入已经按升序给定，无需再排序，只需注意遇到两个式子中相同幂次的，若合并后系数为0注意舍去，不能输出或者增加项的数目。

思路：

创立一个保存幂次和系数结构体用来存储各个项，再用三个结构体数组分别存两个式子和最终答案。

· 代码

```
#include <stdio.h>
#include <stdlib.h>
typedef long long ll;
struct axp
{
    ll a;
    ll p;
};
struct axp arr1[100005];
struct axp arr2[100005];
struct axp ans[200005];
int main()
{
    int n, m;
    int t, i, j, k, idx;
    scanf("%d", &t);
    while(t--){
        scanf("%d", &n);
        for (i = 1; i <= n; i++)
```

```

{
    scanf("%lld", &arr1[i].a);
}
for (i = 1; i <= n; i++)
{
    scanf("%lld", &arr1[i].p);
}
scanf("%d", &m);
for (i = 1; i <= m; i++)
{
    scanf("%lld", &arr2[i].a);
}
for (i = 1; i <= m; i++)
{
    scanf("%lld", &arr2[i].p);
}
i = 1, j = 1, k = 1;
while (i <= n && j <= m)
{
    if (arr1[i].p < arr2[j].p)
    {
        ans[k].a = arr1[i].a;
        ans[k].p = arr1[i].p;
        i++;
    }
    else if (arr2[j].p < arr1[i].p)
    {
        ans[k].a = arr2[j].a;
        ans[k].p = arr2[j].p;
        j++;
    }
    else
    {
        ans[k].a = arr1[i].a + arr2[j].a;
        ans[k].p = arr1[i].p;
        if(ans[k].a == 0){
            k--;
        }
        i++;
        j++;
    }
    k++;
}
if (i <= n) //arr1没遍历完或者arr2没遍历完，都只有一种情况
{
    for (; i <= n; i++)
    {
        ans[k].a = arr1[i].a;
        ans[k].p = arr1[i].p;
        k++;
    }
}
else if (j <= m)
{
    for (; j <= m; j++)
    {
        ans[k].a = arr2[j].a;
        ans[k].p = arr2[j].p;
    }
}

```

```
        k++;
    }
}
printf("%d\n", k-1);
for (idx = 1; idx < k; idx++)
{
    printf("%11d ", ans[idx].a);
}
printf("\n");
for (idx = 1; idx < k; idx++)
{
    printf("%11d ", ans[idx].p);
}
printf("\n");
}
return 0;
}
```