C1-A problem

题目描述

给定整数序列 a1,a2,...,an和 b1,b2,...,bm严格递增的非负整数序列 A1,A2,...,An和 B1,B2,...,Bm，求解如下多项式：

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

数据规模&输入格式

第一行一个正整数 t（1≤t≤5），表示数据组数。

对于每组数据，第一行一个正整数 n（1≤n≤10^5），含义同题目描述。

第二行 n 个整数 a1,a2,...,an（0<|ai|≤10^9），含义同题目描述。

第三行 n 个非负整数 A1,A2,...,An（0≤Ai≤10^9），含义同题目描述。

第四行一个正整数 m（1≤m≤10^5），含义同题目描述。

第五行 n 个整数 b1,b2,...,bm（0<|bi|≤10^9），含义同题目描述。

第六行 n 个非负整数 B1,B2,...,Bm（0≤Bi≤10^9），含义同题目描述。

保证 Ai>Ai−1 对 1<i≤n成立，Bi>Bi−1 对 1<i≤m1 成立，Ai=Bj⇒ai+bj≠0对 1≤i≤n,1≤j≤m 成立。

题解思路

- 首先依据题干，两个多项式的指数序列保证单调递增，因此可以从小指数到大指数依次进行同类项合并。
- 将两个多项式和结果多项式的系数和指数分别存储在数组中。分别用两个指针指向两多项式的第一项。
- 比较两指针所指向的项的指数，若相同，计算系数之和后存入结果数组，两指针均向后移动一项；若不同，则将指数较小的项的系数和指数直接存入结果数组，较小指针向后移一项。
- 重复进行比较操作，直到两指针均指向各自多项式的末尾。若只有一个多项式指针指向末尾，则将另一多项式剩余部分直接复制到结果数组中。

代码

```
##include <stdio.h>

int box[200000] = {0};
int box_index[200000] = {0};
int box_a[100000] = {0};
int box_a_index[100000] = {0};
```

```
int box_b[100000] = {0};
int box_b_index[100000] = {0};
int main() {
    int t = 0;
    scanf("%d", &t);
    while(t--) {
        int n = 0, m = 0;
        int p = 0; //结果数组指针
        int p_a = 0, p_b = 0; //两多项式各自指针
        int i = 0;
        scanf("%d", &n);
        for (i = 0; i < n; i++) {
            scanf("%d", &box_a[i]);
        }
        for (i = 0; i < n; i++) {
            scanf("%d", &box_a_index[i]);
        }
        scanf("%d", &m);
        for (i = 0; i < m; i++) {
            scanf("%d", &box_b[i]);
        }
        for (i = 0; i < m; i++) {
            scanf("%d", &box_b_index[i]);
        }
        //开始进行合并同类项
        while (p_a < n || p_b < m) {
            //两多项式均有余项
            if (p_a < n && p_b < m) {
                if (box_a_index[p_a] < box_b_index[p_b]) {
                    box[p] = box_a[p_a];
                    box_index[p] = box_a_index[p_a];
                    p_a++; p++;
                    continue;
                }
                else if(box_a_index[p_a] > box_b_index[p_b]) {
                    box[p] = box_b[p_b];
                    box_index[p] = box_b_index[p_b];
                    p_b++; p++;
                    continue;
                }
                    //两项指数相同，合并同类项
                else {
                    //合并后该项系数为0
                    if ((box_a[p_a] + box_b[p_b]) == 0) {
                        p_a++; p_b++;
                        continue;
                    }
                    else {
                        box[p] = box_a[p_a] + box_b[p_b];
                        box_index[p] = box_a_index[p_a];
                        p++; p_a++; p_b++;
                        continue;
                    }
                }
            }
```

```
                //若只有a有余项
            else if (p_a < n) {
                box[p] = box_a[p_a];
                box_index[p] = box_a_index[p_a];
                p_a++; p++;
                continue;
            }
                //若只有b有余项
            else {
                box[p] = box_b[p_b];
                box_index[p] = box_b_index[p_b];
                p_b++; p++;
                continue;
            }
        }
        //输出
        printf("%d\n", p);
        for (i = 0; i < p - 1; i++) {
            printf("%d ", box[i]);
        }
        printf("%d\n", box[i]);
        for (i = 0; i < p - 1; i++) {
            printf("%d ", box_index[i]);
        }
        printf("%d\n", box_index[i]);
    }
    return 0;
}
```