

# C1 - F 多项式合并

DeNeRATE 杨佳宇轩

9/18/2024

# 题目描述

- 多组输入
- 输入两个多项式，求多项式合并的结果
- 保证，多项式幂递增且合并后不会出现 0 系数项

## 样例解释

$$\begin{aligned} &1 \times x^1 + 2 \times x^2 + 3 \times x^3 + \\ &\quad 2 \times x^2 + 3 \times x^3 + 4 \times x^4 \\ &= 1 \times x^1 + 4 \times x^2 + 6 \times x^3 + 4 \times x^4 \end{aligned}$$

# 数据范围

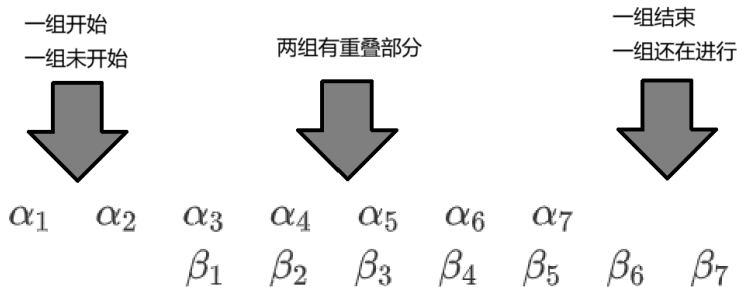
- $1 \leq n, m \leq 10^5 \Rightarrow$  线性时间复杂度
- $0 \leq a_i, b_i, A_i, B_i \leq 10^9 \Rightarrow$  多项式相加后不超 int

# 数据范围

- $1 \leq n, m \leq 10^5 \Rightarrow$  线性时间复杂度
- $0 \leq a_i, b_i, A_i, B_i \leq 10^9 \Rightarrow$  多项式相加后不超 `int`

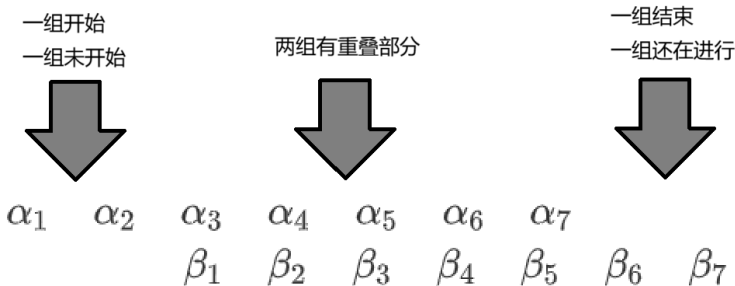
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 采用双指针的思想，两个多项式分别存入两个 vector 中，按幂从头开始向后移动，直至两个多项式都以合并完毕
- 会出现三种情况，一组的幂小于另一组、两组幂相同、一组已经排完，另一组还在进行



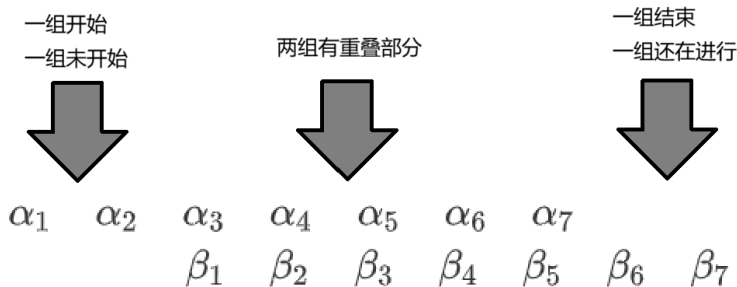
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 采用双指针的思想，两个多项式分别存入两个 vector 中，按幂从头开始向后移动，直至两个多项式都以合并完毕
- 会出现三种情况，一组的幂小于另一组、两组幂相同、一组已经排完，另一组还在进行



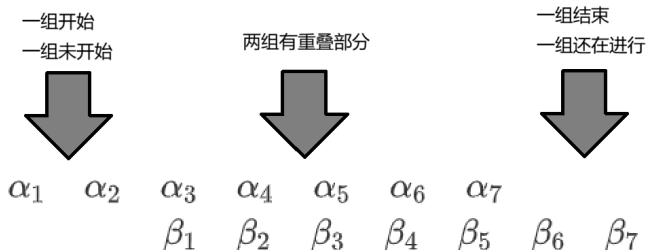
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 采用双指针的思想，两个多项式分别存入两个 vector 中，按幂从头开始向后移动，直至两个多项式都以合并完毕
- 会出现三种情况，一组的幂小于另一组、两组幂相同、一组已经排完，另一组还在进行



# 实现思路

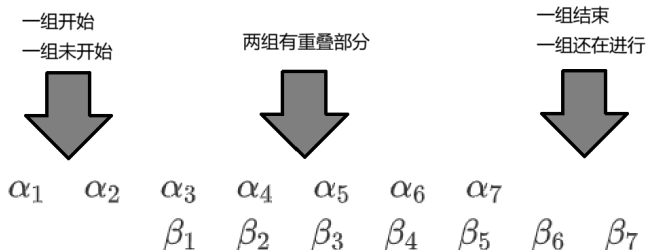
- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 故我们对于当前双指针的状态，
- 讨论其是否是有一组已结束，若已结束，则仅在答案数组中放入另一组的项
- 否则若一组的幂小于另一组，则放该项
- 否则，将  $\alpha$  和  $\beta$  组的系数加起来存入答案





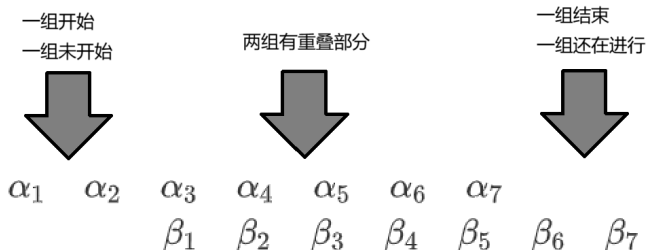
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 故我们对于当前双指针的状态，
- 讨论其是否是有一组已结束，若已结束，则仅在答案数组中放入另一组的项
- 否则若一组的幂小于另一组，则放该项
- 否则，将  $\alpha$  和  $\beta$  组的系数加起来存入答案



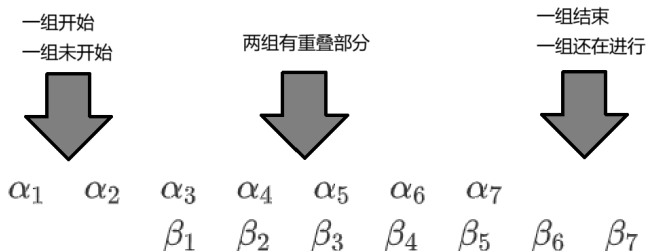
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 故我们对于当前双指针的状态，
- 讨论其是否是有一组已结束，若已结束，则仅在答案数组中放入另一组的项
- 否则若一组的幂小于另一组，则放该项
- 否则，将  $\alpha$  和  $\beta$  组的系数加起来存入答案



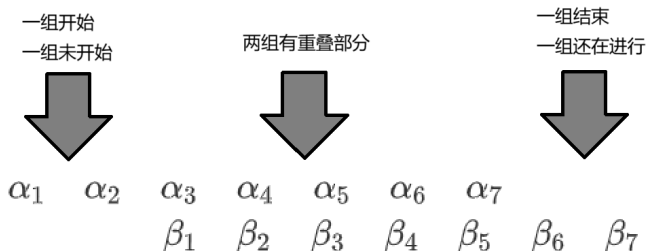
# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 故我们对于当前双指针的状态，
- 讨论其是否是有一组已结束，若已结束，则仅在答案数组中放入另一组的项
- 否则若一组的幂小于另一组，则放该组项
- 否则，将  $\alpha$  和  $\beta$  组的系数加起来存入答案



# 实现思路

- 将两个多项式分别称为  $\alpha$  和  $\beta$
- 故我们对于当前双指针的状态，
- 讨论其是否是有一组已结束，若已结束，则仅在答案数组中放入另一组的项
- 否则若一组的幂小于另一组，则放该组项
- 否则，将  $\alpha$  和  $\beta$  组的系数加起来存入答案



# 代码示例

```
1  int posa = 1, posb = 1;
2  vector<pair<int, int>> ans;
3  while(posa <= n || posb <= m) {
4      if(posa > n)
5          ans.push_back({b[posb], B[posb]}), posb++;
6      else if(posb > m)
7          ans.push_back({a[posa], A[posa]}), posa++;
8      else if(A[posa] < B[posb])
9          ans.push_back({a[posa], A[posa]}), posa++;
10     else if(A[posa] > B[posb])
11         ans.push_back({b[posb], B[posb]}), posb++;
12     else
13         ans.push_back({a[posa] + b[posb], A[posa]}), posa++,
14         posb++;
15 }
```

Thank you!