

Algorithm 2-1

逆序數對

Description:

Inversion number is the number of pairs of inverted numbers in a series. It's often used to indicate the extent of unsortedness. We define a pair of numbers is a pair of inverted numbers if $i < j$ and $a_i > a_j$.

Input Format:

第一行為數列 a 的大小，第二行為組成 a 的 n 個整數

First line is the size of a . Second line contains the numbers in a .

- $2 \leq n \leq 10^6$
- $-10^8 \leq a_i \leq 10^8$

Output the inversion number of a . The number may be big. Please mod it with 524287 before output. (Use merge sort)

Sample input	Sample Output 1
4 4 3 2 1	6
6 4 5 10 4 7 1	8

Algorithm 2-2

大樂透 (Use recursion)

Description:

Arashi87 picked up a red envelope on the street during the Lunar New Year. He decides to use the money inside it to buy lottery. Here we need six numbers for the lottery. Arashi87 have a strange habit for picking numbers. He will lists all combinations on a big white paper and hang it on a wall. He then shoots it with darts randomly to pick which combination of numbers he will buy for the lottery. However, there are just too many combinations to pick 6 numbers out of 49 numbers. Therefore, Arashi87 picks 7~12 numbers from 1~49 first. Your job here is to list all the combinations from the numbers Arashi87 picked.

Input Format:

You will be given an integer k ($6 < k < 13$) first. The integers following k is the number Arashi87 picked, and they are listed in ascending order.

Output Format:

Print all the possible combinations. (One combination for each line) Each combination should be in ascending order. There should be a space between each number. All combination must be printed in dictionary order a.k.a Lexicographic order. This means sorting the combination according to the smallest number first. If the smallest number are the same, use the second smallest number to sort, and so on.

Sample input	Sample Output 1
7 1 2 3 4 5 6 7	1 2 3 4 5 6

	1 2 3 4 5 7 1 2 3 4 6 7 1 2 3 5 6 7 1 2 4 5 6 7 1 3 4 5 6 7 2 3 4 5 6 7
8 1 2 3 5 8 13 21 34	1 2 3 5 8 13 1 2 3 5 8 21 1 2 3 5 8 34 1 2 3 5 13 21 1 2 3 5 13 34 1 2 3 5 21 34 1 2 3 8 13 21 1 2 3 8 13 34 1 2 3 8 21 34 1 2 3 13 21 34 1 2 5 8 13 21 1 2 5 8 13 34 1 2 5 8 21 34 1 2 5 13 21 34 1 2 8 13 21 34 1 3 5 8 13 21 1 3 5 8 13 34 1 3 5 8 21 34 1 3 5 13 21 34 1 3 8 13 21 34 1 5 8 13 21 34 2 3 5 8 13 21 2 3 5 8 13 34 2 3 5 8 21 34 2 3 5 13 21 34 2 3 8 13 21 34 2 5 8 13 21 34 3 5 8 13 21 34

Algorithm 2-3

2D Rank Finding

Description:

In the XY-plane, we say a point **dominates** another point if and only if $x_1 > x_2$ and $y_1 > y_2$.

The **rank** of a point is the number of points which are dominated by the point.

Given a set of points in the XY-plane, find the **rank** of each point.

No two points share the same coordinates.

Input Format:

Input starts with an integer n , where n denotes the number of points.

Each of the next n lines gives two integers which are the x and y coordinates of a point.

- $5 \leq n \leq 3 \cdot 10^5$
- $-2^{31} \leq x, y \leq 2^{31}$

Sample input	Sample Output 1
5 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4
7 3 1 4 1 5 9 2 6 5 3 5 8 9 7	0 1 5 0 2 4 4

