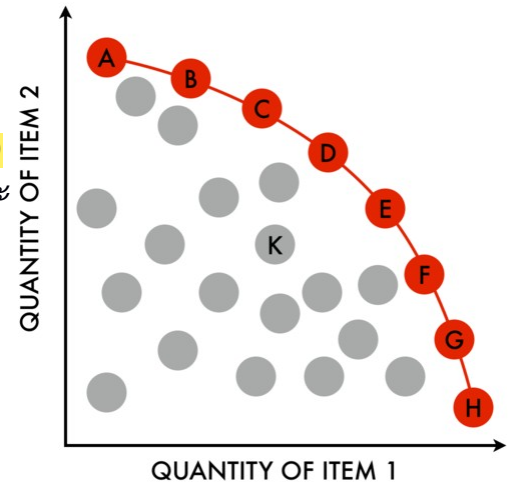


Pareto Frontier

Time limit: 1 sec

โจทย์ข้อนี้จะเป็นการคำนวณหา Pareto Frontier ของข้อมูลจุดใน 2 มิติ โดย Pareto Frontier มีนิยามดังต่อไปนี้
กำหนดให้มีเซตของจุด 2 มิติ $A = \{a_0, a_1, \dots, a_{(n-1)}\}$
โดยที่แต่ละจุดจะถูกระบุกำกับด้วยหมายเลข 1 ถึง n และแต่ละจุดมีพิกัด x, y เป็น $a_i.x$ และ $a_i.y$ เราจะเรียกจุด a_i ว่า "ถูกครอบงำโดย" จุด a_j ก็ต่อเมื่อ $a_i.x \leq a_j.x$ และ $a_i.y \leq a_j.y$
กำหนดให้ Pareto Frontier ของเซต A คือ เซตของ A ที่ประกอบด้วยสมาชิกที่ไม่ถูกครอบงำโดยจุดอื่นใดเลย
ตัวอย่างเช่นในรูปด้านขวามือนี้ จุดสีแดงทั้งหมดคือจุดที่เป็น Pareto Frontier



จงเขียนโปรแกรมที่คำนวณ Pareto Frontier จากเซต A ที่กำหนดให้

Input

- บรรทัดแรกประกอบด้วยจำนวนเต็ม N ซึ่งระบุจำนวนจุดในเซต A ทั้งหมด ($1 \leq N \leq 100,000$)
- อีก N บรรทัดถัดมาจะระบุจุดแต่ละจุด บรรทัดละ 1 จุด แต่ละบรรทัดประกอบด้วยตัวเลขสองตัว ซึ่งระบุพิกัดในแกน x และ y ของจุดดังกล่าว ($0 \leq x, y \leq 100,000$) รับประกันว่าไม่มีสองจุดใด ๆ ที่มีพิกัด x เหมือนกันเลย

Output

ประกอบด้วยบรรทัดหนึ่งบรรทัด โดยให้ระบุจำนวนของจุดที่เป็น Pareto Frontier

คำแนะนำ

- มีข้อมูลทดสอบจำนวน 50% ที่มีค่า N ไม่เกิน 1000 ซึ่งสามารถหาคำตอบได้ด้วยวิธีการ $O(N^2)$
- มีโค้ดเริ่มต้นที่เป็น Divide & Conquer ให้แล้ว โดยมีฟังก์ชัน `vector<int> pareto(int l, int r)` ซึ่งจะคำนวณ Pareto frontier ของ $A[l]$ ถึง $A[r]$ โดยคืน vector ของหมายเลขของจุดที่เป็น Pareto frontier ของ $A[l]$ ถึง $A[r]$ ซึ่งมีหลักการทำงานดังต่อไปนี้
 - แบ่ง P ออกเป็นสองส่วนย่อย โดยที่แบ่งตามค่าพิกัด X กล่าวคือ ให้ $m = (l+r) / 2$ ปัญหาย่อยแรกคือจุดหมายเลข l ถึง m และปัญหาย่อยที่สองคือจุดหมายเลข m+1 ถึง r
 - เรียก Pareto แบบ recursive เพื่อแก้ปัญหาย่อยทั้งสองส่วน
 - ทำการรวม Pareto Frontier จากทั้งสองปัญหาย่อยมาเป็น Pareto Frontier ของปัญหาหลัก

- นิสิตจะต้องเขียนโปรแกรมในขั้นตอนการรวมผลเอง
- นิสิตไม่จำเป็นจะต้องใช้โค้ดเริ่มต้นดังกล่าวก็ได้

ตัวอย่าง

Input	Output
3 50 10 30 30 10 50	3 // ทุกจุดเป็น Pareto frontier
3 10 10 20 20 30 30	1 // เฉพาะจุด 30 30 เท่านั้นที่เป็น Pareto frontier
4 20 40 21 21 40 20 41 41	1 // จุดที่อยู่ใน Pareto Frontier คือ 41 41 เท่านั้น