

K-ary Lowest Common Ancestor

(1 sec, 512mb)

(สองย่อหน้าถัดไปนี้อธิบายถึง K-ary heap ถ้าเข้าใจ K-ary heap อยู่แล้วข้ามไปได้เลย)

คลาส CP::priority_queue นั้นใช้โครงสร้างข้อมูลที่ชื่อว่า binary heap ซึ่งเป็น full binary tree แบบหนึ่ง กล่าวคือ ปมหนึ่งปมในต้นไม้จะมีลูกไม่เกิน 2 ลูก และทุก ๆ ชั้นความสูงของต้นไม้มีปมเต็มทุกชั้น ยกเว้นเฉพาะชั้นที่ลึกที่สุดที่ปมอาจจะไม่เต็มได้ แต่ปมทุกปมต้องอยู่เต็มจากซ้ายไปขวา จากข้อกำหนดดังกล่าวทำให้ binary heap มีความสูงเป็น $O(\log_2 N)$ เมื่อ N คือจำนวนข้อมูลใน binary heap ซึ่งความสูงของต้นไม้ก็คือเวลาที่ binary heap ใช้ในการทำงาน push และ pop

เราสามารถทำให้ heap นั้นมีความสูงน้อยลงได้โดยการเปลี่ยนไปใช้ K-ary heap แทน โดยที่ K-ary Heap นั้นคือ Heap ที่เป็นต้นไม้ที่มีจำนวนลูกไม่เกิน k ลูก โดยที่กฎต่าง ๆ ของ binary heap นั้นยังคงใช้กับ k-ary heap เหมือนเดิม กล่าวคือ ทุก ๆ ชั้นความสูงของต้นไม้จะต้องมีปมเต็มทั้งชั้นในทุก ๆ ชั้น ยกเว้นชั้นที่ลึกที่สุดที่อาจจะมีปมไม่เต็มทั้งชั้นก็เป็นได้ แต่ปมทุกปมในชั้นดังกล่าวจะต้องอยู่เต็มจากซ้ายไปขวา การใช้ K-ary Heap นั้นจะทำให้ได้ต้นไม้ซึ่งมีความสูงเป็น $O(\log_k N)$ โดยยังสามารถเก็บข้อมูลโดยใช้ dynamic array ได้ตามปกติ

(ตั้งแต่ย่อหน้านี้เป็นต้นไป อธิบายถึงโจทย์)

เรากำหนดให้ “ปมบรรพบุรุษ” ของปมหมายเลข p ใน K-ary heap ขนาด N ปมนั้น คือปมทุกปมที่อยู่ใน path จากปมรากถึงปม p (รวมปมรากด้วย)

ในโจทย์ข้อนี้เราต้องการหา “ปมบรรพบุรุษร่วมที่อยู่ลึกที่สุด” (Lowest Common Ancestor) ของปม p และ q ที่กำหนดให้ โดยกำหนดให้ “ปมบรรพบุรุษร่วม ของ p และ q ” คือปมใด ๆ ที่เป็นปมบรรพบุรุษของทั้งปม p และ ปม q (ให้สังเกตว่า สำหรับคู่ปม p, q ใด ๆ ในต้นไม้ นั้นย่อมมีปมบรรพบุรุษร่วมอยู่เสมอ)

ให้ปมบรรพบุรุษร่วมที่อยู่ลึกที่สุดก็คือ ปมบรรพบุรุษที่มีหมายเลขปมมากที่สุด

ข้อมูลนำเข้า

- บรรทัดแรกประกอบด้วยจำนวนเต็ม 2 จำนวนได้แก่ค่า N และ K ซึ่งระบุ K-Ary Heap ขนาด N ปมที่เราสนใจ ($2 \leq K < N \leq 1,000,000,000$)
- บรรทัดที่สองประกอบด้วยจำนวนเต็มสองจำนวนคือ p และ q ($0 \leq p, q < N$) ซึ่งระบุหมายเลขปมที่เราสนใจ

ข้อมูลส่งออก

มีทั้งหมด 1 บรรทัดซึ่งระบุหมายเลขปมบรรพบุรุษร่วมที่อยู่ลึกที่สุดของปม p และ q

ชุดข้อมูลทดสอบ

- 5% $N = 3, K = 2$ (เป็น binary heap ที่มี 2 ชั้นเสมอ)
- 10% $N = 7, K = 2$ (เป็น binary heap ที่มี 3 ชั้นเสมอ)
- 15% $N \leq 15, K = 2$ (เป็น binary heap ที่มีไม่เกิน 4 ชั้น)
- 15% K-ary heap นี้มีไม่เกิน 3 ชั้น
- 20% ปมบรรพบุรุษร่วมของ p และ q จะอยู่ห่างจาก p และ q ไม่เกิน 2 ชั้น
- 35% ไม่มีเงื่อนไขอื่นใด

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
7 2 5 6	2
11 2 4 9	4 // บรรพบุรุษของปมที่ 9 คือ ปมที่ 4 ปมที่ 1 และปมที่ 0 ตามลำดับ
14 2 2 13	2 // บรรพบุรุษของปมที่ 13 คือ ปมที่ 6 ปมที่ 2 และปมที่ 0 ตามลำดับ
9 2 8 1	1 // บรรพบุรุษของปมที่ 8 คือ ปมที่ 3 ปมที่ 1 และปมที่ 0 ตามลำดับ
21 4 13 16	3
15 4 7 0	0
18 4 3 6	0
730369621 13 85582317 506383	2996

คำแนะนำ

เราสามารถทำข้อนี้ได้โดยหารายการของปมบรรพบุรุษทั้งหมดของ p และ q โดยเรียงจากปมที่อยู่ "ใกล้" ปม p (และ q) ไปยังปมที่อยู่ "ไกล" แล้วให้สังเกตว่า รายการปมบรรพบุรุษนี้จะมีช่วงท้ายที่เหมือนกัน ปมบรรพบุรุษร่วมก็คือปมที่อยู่ต้นรายการมากที่สุดที่ซ้ำกัน