

K-ary Path To Node

(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 ยกเว้นชั้นที่ลึกที่สุดที่อาจจะไม่เต็มทั้งชั้นก็เป็นได้ แต่ปมทุกปมในชั้นดังกล่าวจะต้องอยู่เต็มจากซ้ายไปขวา การใช้ K-ary Heap นั้นจะทำให้ได้ต้นไม้ซึ่งมีความสูงเป็น $O(\log_K N)$ โดยยังสามารถเก็บข้อมูลโดยใช้ dynamic array ได้ตามปกติ

(ตั้งแต่ย่อหน้านี้เป็นต้นไป อธิบายถึงโจทย์)

โจทย์ข้อนี้จะกำหนดหมายเลขปม p ของ K-ary Heap มาให้ จงเขียนโปรแกรมเพื่อคำนวณว่า หากเราต้องการเดินทางจากปมราก (ปมหมายเลข 0) ไปยังปมที่กำหนดให้ นั้น โดยผ่านเส้นเชื่อมต่าง ๆ ของต้นไม้ เราจะต้องเดินทางอย่างไร

การระบุการเดินทางนั้นให้ระบุเป็นขั้นตอน แต่ละขั้นคือการระบุจากปมปัจจุบันที่อยู่ ณ เราเลือกเดินไปยังปมลูกปมใด โดยในขั้นตอนแรกสุดนั้นให้ถือว่าเราเริ่มต้นที่ปมราก และเมื่อจบการเดินทางแล้ว เราจะต้องอยู่ที่ปมที่ต้องการพอดี

ข้อมูลนำเข้า

- บรรทัดแรกประกอบด้วยจำนวนเต็ม 2 จำนวน ได้แก่ค่า N และ K ซึ่งระบุจำนวนปมและค่า K ของ K-Ary Heap ที่เราสนใจ ($1 \leq N, K \leq 10^{18}$)
- บรรทัดที่สองประกอบด้วยจำนวนเต็มหนึ่งจำนวนคือ p ($0 \leq p < N$)

ข้อมูลส่งออก

มีไม่เกิน 2 บรรทัดซึ่งระบุการเดินทางตามรูปแบบดังต่อไปนี้

- บรรทัดแรกระบุจำนวนขั้นตอนในการเดินทาง (หากปม p คือปมรากให้ถือว่าจำนวนขั้นตอนในการเดินทางคือ 0 เนื่องจากไม่ต้องเดินทางใด ๆ) **รับประกันว่าจะมีจำนวนขั้นตอนในการเดินทางไม่เกิน 1,000,000**
- บรรทัดถัดมาประกอบด้วยจำนวนเต็ม ที่มีจำนวนเท่ากับค่าที่ระบุในบรรทัดแรก ซึ่งระบุแต่ละขั้นตอนในการเดินทางตามลำดับ ค่าแต่ละค่าจะระบุไว้ในชั้นดังกล่าวเราเลือกเดินทางไปยังลูกลำดับที่เท่าไร (ให้ลูกซ้ายสุดมีลำดับเป็น 0 และลูกขวาสุดมีลำดับเป็น $K-1$)

หมายเหตุ

เนื่องจากคำตอบอาจมีขนาดใหญ่มาก ดังนั้นแนะนำให้ใช้ long long แทน int

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 ที่มีไม่เกิน 3 ชั้น
- 15% K-ary heap ที่มีไม่เกิน 4 ชั้น
- 20% N , $K \leq 100,000,000$
- 20% ไม่มีเงื่อนไขอื่นใด

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
3 2 1	1 0
7 2 5	2 1 0
14 2 12	3 1 0 1
225 15 189	2 11 8
11111111 10 10101010	7 8 9 8 9 8 9 9
1606214373614 9724 641628575319	3 6784 6675 3734