

K-ary Heap Check

(1 sec, 512 mb)

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

คลาส `CP::priority_queue` นั้นใช้โครงสร้างข้อมูลที่ว่า binary heap ซึ่งเป็น full binary tree แบบหนึ่ง กล่าวคือ ปมหนึ่งปมในต้นไม้จะมีลูกไม่เกิน 2 ลูก และปมใด ๆ จะมีค่ามากกว่าปมลูกของปมนั้นเสมอ และทุก ๆ ชั้นความสูงของต้นไม้จะมีปมเต็มทุกชั้น 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 ปมใด ๆ จะมีค่ามากกว่าปมลูกของปมนั้นเสมอ และทุก ๆ ชั้นความสูงของต้นไม้จะต้องมีปมเต็มทั้งชั้นในทุก ๆ ชั้น ยกเว้นชั้นที่ลึกที่สุดที่อาจจะไม่มีปมเต็มทั้งชั้นก็เป็นได้ แต่ปมทุกปมในชั้นดังกล่าวจะต้องอยู่เต็มจากซ้ายไปขวา การใช้ K-ary Heap นั้นจะทำให้ได้ต้นไม้ซึ่งมีความสูงเป็น $O(\log_k N)$ โดยยังสามารถเก็บข้อมูลโดยใช้ dynamic array ได้ตามปกติ

ในโจทย์ข้อนี้เราต้องการตรวจสอบว่า ต้นไม้แบบ full k-ary tree (คือต้นไม้ที่มีลูกไม่เกิน k ปมที่เต็มทุกชั้น ยกเว้นชั้นล่างสุดที่จะเต็มจากซ้ายไปขวา) นั้นมีลักษณะตรงตามกฎของ K-ary heap หรือไม่ โดย array ที่ให้มาจะเป็นค่าของปมต่าง ๆ ที่ใช้วิธีการเก็บข้อมูลในรูปแบบเดียวกับ `CP::priority_queue` แต่พิจารณาเป็นแบบ k-ary heap แทน กล่าวคือ ข้อมูลแต่ละช่องใน array นี้จะเป็นข้อมูลของ ต้นไม้แบบ full k-ary tree โดยเป็นข้อมูลตั้งแต่ชั้นบนสุด (ความลึก 0) จนถึงชั้นล่างสุดของต้นไม้ดังกล่าว และข้อมูลในแต่ละชั้นนั้นจะเป็นค่าของปมที่อยู่ซ้ายสุดจนถึงขวาสุดตามลำดับ

ในโจทย์ข้อนี้ นิสิตจะต้องรับข้อมูล array ดังกล่าวจำนวนหลาย ๆ ชุด และในแต่ละชุดให้ตอบว่า array ดังกล่าวแสดงถึง k-ary heap หรือไม่

ข้อมูลนำเข้า

- บรรทัดแรกประกอบด้วยจำนวนเต็ม m ซึ่งระบุถึงจำนวน array ที่ต้องการตรวจสอบ
- หลังจากนั้นอีก $2m$ บรรทัด จะเป็นข้อมูลของ array ที่อธิบายถึง k-ary tree แต่ละต้น ต้นละสองบรรทัดตามรูปแบบต่อไปนี้
 - บรรทัดแรกประกอบด้วยจำนวนเต็มสองตัวคือ n และ k ซึ่งระบุถึงจำนวนปมในต้นไม้ และ ค่า k (จำนวนลูกมากที่สุดของแต่ละปม) โดยที่
$$1 \leq n, k \leq 50,000$$
 - บรรทัดที่สองประกอบด้วยจำนวนเต็ม n ตัว ซึ่งระบุถึงค่าของปมต่าง ๆ ตั้งแต่ชั้นบนสุดถึงชั้นล่างสุด โดยที่แต่ละชั้นระบุค่าของปมซ้ายสุดถึงปมขวาสุดของชั้นดังกล่าว (รับประกันว่า ค่าของปมต่าง ๆ จะไม่ซ้ำกันเลย)

ข้อมูลส่งออก

- ต้องมีทั้งหมด m บรรทัด แต่ละบรรทัดระบุค่า true หรือ false เพื่อบอกว่า array ที่ตรวจสอบนั้นเป็น k-ary heap หรือไม่ตามลำดับ

ชุดข้อมูลทดสอบ

- 20% $n \leq 100$, $k = 2$
- 20% $k = 3$
- 20% $1 \leq k \leq 4$
- 40% ไม่มีเงื่อนไขอื่นใด

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
3	true
1 1	true
10	false
7 2	
10 9 4 7 8 1 2	
7 2	
10 9 4 1 2 7 8	
1	true
12 3	
12 11 10 9 8 7 6 5 4 3 2 1	

*รับประกันว่าทุกชุดทดสอบ $M \leq 1000$ *

หมายเหตุ

ข้อมูลนำเข้าและส่งออกของโปรแกรมนี้มีเป็นจำนวนมาก การทำงานตามปกติของ cin และ cout นั้นช้าเกินไป ขอให้เรียกคำสั่งดังต่อไปนี้เป็นคำสั่งแรกใน main function เพื่อเพิ่มความเร็วให้กับ cin และ cout

```
std::ios_base::sync_with_stdio(false); std::cin.tie(0);
```