

## Height of K-Ary Heap

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  $O(\log_k N)$

จงเขียนโปรแกรมเพื่อคำนวณความสูงของต้นไม้ K-ary Heap ที่มีข้อมูลจำนวน  $n$  ข้อมูล

### ข้อมูลนำเข้า

มีหนึ่งบรรทัดประกอบด้วยจำนวนเต็ม  $n$  และ  $K$  ซึ่งระบุจำนวนข้อมูลที่ใส่เข้าไปใน K-ary Heap โดยที่  $1 \leq n \leq 2 \cdot 10^9$  และ  $1 \leq K \leq 100$

### ข้อมูลส่งออก

มีหนึ่งบรรทัดประกอบด้วยจำนวนเต็มหนึ่งตัว ซึ่งระบุความสูงของต้นไม้ K-ary Heap

### ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
1 2	0
7 2	2
8 2	3
21 4	2
22 4	3
1000000000 1	999999999
1997023749 62	6

### ข้อควรระวัง

ขนาดของต้นไม้อาจจะใหญ่มาก แนะนำให้ใช้ตัวแปรประเภท long long (ตัวอย่างเช่น แทนที่จะประกาศตัวแปรชื่อ  $x$  ด้วย `int x` ก็เปลี่ยนไปประกาศด้วย `long long x` แทน จะทำให้  $x$  สามารถเก็บค่าขนาดใหญ่ได้)