**SwiftToTwo**

เรามาลองคิดดูก่อนว่า เลขไม่เกิน 1017 จะมีจำนวนเต็มที่บวกตัวแรกที่หารไม่ลงได้มากสุดเท่าไหร่

สังเกตว่าการที่จำนวนเฉพาะหารเลขๆนึงลงได้ เลขๆนั้นต้องมีจำนวนเฉพาะนั้นคูณอยู่ด้วย

เลขแรกที่หารด้วย 2 ลงคือ 2

เลขแรกที่หารด้วย 2,3 ลงคือ 2x3

เลขแรกที่หารด้วย 2,3,5 ลงคือ 2x3x5

เลขแรกที่หารด้วย 2,3,5,7 ลงคือ 2x3x5x7

….

เลขแรกที่หารด้วย 2,3,5,7,….,47 ลงคือ 2x3x5x…x37x41x43x47 ซึ่ง >1017

ดังนั้น เราสามารถประมาณได้เลยว่า เลขทุกตัวที่ <=1017 จะมีเลขตัวแรกที่หารไม่ลง <47 แน่นอน

ทีนี้ เราจะมีหาว่า 1,2,3,…,a มีเลขที่มี x เป็นตัวแรกที่หารไม่ลงกี่ตัวกัน

ก่อนอื่นเราจะมาหาเลขตัวแรกที่สามารถหารด้วย 1,2,3,…,x ลงตัว

ให้ dp[i] = เลขตัวแรกที่สามารถหารด้วย 1,2,3,…,i ลงตัว

ทีนี้เรามาดูว่า ถ้าเรารู้เลขตัวแรกที่สามารถหารด้วย 1,2,3,…,i-1 ลงตัว เราจะหาเลขตัวแรกที่หารด้วย 1,2,3,…,i ยังไง

สังเกตว่า เราก็แค่เอา dp[i-1] มาคูณอะไรซักอย่างให้หน่อยที่สุดให้มันหารด้วย i ลงให้ได้ซะ

ซึ่งมันก็คือส่วนที่ไม่อยู่ใน หรม ของ dp[i-1] กับ i ของ i นั่นเอง

หรือก็คือ i/gcd(dp[i-1],i)

จะได้ว่า dp[i] = dp[i-1] \* (i/gcd(dp[i-1],i)