**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] มาคูณอะไรซักอย่างให้น้อยที่สุดให้มันหารด้วย 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)

ทีนี้ เราต้องการหาว่า มีเลขกี่ตัวใน 2,3,….,a ที่มี i+1 เป็นตัวแรกที่หารไม่ลงตัว

สังเกตว่า dp[i] คือเลขที่น้อยที่สุดที่หารด้วย 1,2,3,…,i ลงตัว

สังเกตว่า dp[i] หารด้วย 1,2,3,…,i ลงตัว แล้วพวกพหุคูณของ dp[i] ก็ต้องเป็นแบบเดียวกันแน่นอน

ดังนั้นในเลข 1 ถึง a จากมีเลขที่หารด้วย 1,2,3,…,i ลงเท่ากับ a/dp[i] (แต่อาจจะหารด้วย i+1 ลงหรือไม่ลงก็ได้)

แต่เราอยากได้เฉพาะเลขที่หารด้วย i+1 ไม่ลง เราก็ลบออกด้วยจำนวนเลขที่หารด้วย 1,2,3,…,i+1 ลงซะเลย!

ดังนั้น จำนวนเลขที่หารด้วย 1,2,3,…,i ลงตัวแล้วหารด้วย i+1 ไม่ลงจะเท่ากับ a/dp[i]-a/dp[i+1]

ทีนี้ สังเกตว่า เลขทุกตัวที่หารด้วย i+1 ไม่ลงเป็นตัวแรกจะมีรูปแบบการทำ swift คล้ายกันเลยคือ

เลขนั้น -> i+1 -> … -> 2

จะเป็นรูปแบบนี้เสมอ

ดังนั้น **หมายความว่า เลขที่ทุกที่มีตัวแรกหารด้วย i+1 ไม่ลง จะมีความยาวในการทำ swift เท่ากัน!!!!**

**และความยาวในการทำ swift เลขนั้นจะเท่ากับ “1+ความยาวการทำ swift ของเลข i+1”**

ซึ่งเราก็หาความยาวของการทำ swift ของเลข 2,3,..,41 ก็พอ

ทีนี้ เรารู้แล้วว่า

1. เลขที่มี i+1 เป็นตัวแรกที่หารไม่ลงตัวจะมีจำนวนเท่ากับ a/dp[i]-a/dp[i+1]

2. ความยาวในการทำ swift ของเลขที่มีตัวเลขที่หารไม่ลงเป็น i+1 จะเท่ากับ **1+ความยาวการทำ swift ของ i+1**

3. เลขที่มีเลขแรกที่แรกที่หารไม่ลงเท่ากัน จะมีความยาวในการทำ swift เท่ากัน

ดังนั้น เราก็ไล่หาจำนวนที่มีตัวแรกที่หารไม่ลงตัวเป็น i+1 เอาจำนวนนั้นมาคูณด้วยความยาวในการทำ swift เลขพวกนั้น แล้วบวกใส่คำตอบ หรือก็คือ

เท่านี้เราก็จะสามารถหา swift(3)+swift(4)+…+swift(a) ได้แล้ววววววว