

Lowest Common Multiple

ปัญหา

- กำหนด 2 จำนวน A และ B มาให้จงหาตัวคูณร่วมน้อย (Lowest Common Multiple(LCM))
- เราพยายามหา LCM ของ A และ B
- LCM คืออะไร
- มันคือจำนวนบวกที่น้อยที่สุดที่ A และ B หารลงตัว

เราจะหามันอย่างไร

- จริงๆ มันมีสูตรพื้นฐานอยู่คือ
- $LCM(A, B) \times GCD(A, B) = A \times B$
- นั่นคือ $LCM(A, B) = \frac{A \times B}{GCD(A, B)}$
- ตัวอย่างเช่น
- $LCM(6, 15) = \frac{6 \times 15}{GCD(6, 15)} = \frac{90}{3} = 30$

ข้อควรระวังตอน code

- เมื่อเราแปลงสมการก่อนหน้าในการหา LCM ถ้าแปลงตรงๆ จะได้
ประมาณนี้

```
int lcm ( int a, int b ) {  
    return ( a * b ) / gcd ( a, b );  
}
```

- มันก็ดูปกติดี แต่เห็นอะไรไม่เข้าท่าไหม

- Code ข้างต้นอาจจะเกิด overflow ได้
- ตัวอย่างเช่น $\text{LCM}(2^{20}, 2^{15})$ จากที่ดูเราพบว่า $\text{LCM}(2^{20}, 2^{15})$ มีค่า 2^{20}
- แต่ code ของเราทำงานคือ มันจะพยายาม เอา $2^{20} \times 2^{15}$ ก่อนซึ่งจะได้ 2^{35} ซึ่งใหญ่กว่า int แล้วก็จะ overflow ทำให้ได้คำตอบผิด

- ทางแก้ไม่ยาก

- $$LCM(A, B) = \frac{A \times B}{GCD(A, B)} = \frac{A}{GCD(A, B)} \times B$$

- นั่นคือ

```
int lcm ( int a, int b ) {  
    return ( a / gcd ( a, b ) ) * b;  
}
```