Euclidean Algorithm

Greatest common divisor

ปัญหา

- กำหนดจำนวน A และ B มาให้ จงหาจำนวนที่มากที่สุดที่หาร A และ B ลงตัว หรือตัวหารร่วมมากนั่นเอง (Greatest Common Divisor(GCD))
- ส่วนจำนวนสองจำนวนนั้นจะถูกเรียกว่า **จำนวนเฉพาะสัมพัทธ์**
- ตัวอย่างเช่น GCD(2,8)=2, GCD(3,4)=1 , GCD(12,15)=3

- ตัวหารร่วมมากมีประโยชน์ในการทำเศษส่วนอย่างต่ำ ตัวอย่างเช่น
- $\frac{42}{56} = \frac{3*14}{4*14} = \frac{3}{4}$

เราจะหามันได้อย่างไร

- มีหลายวิธีในการหา GCD
- หนึ่งในนั้นคือ list รายการตัวที่หาร A และ B ได้จากนั้นหาตัวเหมือนกันที่มากที่สุดจาก 2 list นั้น ซึ่งเป็นวิธีธรรมดา และใช้เวลานาน

- อีกวิธีคือการใช้ Euclidean Algorithm การทำงานก็ไม่ยาก
- GCD(a, b) = GCD(b, a%b)
- เนื่องจาก GCD(b, a%b) มีสถานะที่เล็กลง มันจึงหาค่าง่ายกว่าค่าเริ่มต้น
- และเราจะใช้หลักการนี้ในการทำให้สถานะเล็กลงเรื่อยๆ จนกระทั่งคำตอบเห็นได้ชัดซึ่งสถานะที่เห็นได้ชัดคือ GCD(a, a) = a และ GCD(a, 0)=a นั่นเอง

Euclidean Algorithm

- Recursive version
- เขียนสั้นๆ โดยใช้ หลักการเมื่อกี้เลย

```
int gcd(int a, int b){
      if(b==0) return a;
      return gcd(b,a%b);
}
```

- Iterative version
- วิธีนี้ก็จะลด overhead ของ recursion และทำให้ code รันเร็วขึ้น

```
int gcd(int a, int b){
    while(b){
        a = a%b;
        swap(a,b);
    }
    return a;
}
```

- อย่างไรก็ตาม ใน C++ นั้นมี building function ให้ใช้ด้วยคือ
- `__gcd(a,b)`
- ลองดู

พิสูจน์

- เราได้ใช้สมการหนึ่งใน code เรานั่นคือ GCD(a, b) = GCD(b, a%b)
- เราต้องการพิสูจน์ว่ามันถูกต้อง

- กำหนดให้ g = GCD(a, b) นั่นคือ a = k x b + r เมื่อ k เป็น non-negative integer และ r เป็นเศษเหลือ
- เนื่องจาก g หาร a ลงตัว แสดงว่า g หาร k x b + r ลงตัวด้วย และ เนื่องจาก g หาร b ลงตัว ดังนั้น g ก็หาร k x b ได้ลงตัว ทำให้ได้ว่า g ต้องการ r ลงตัว ไม่ใช่นั้น g จะหาร k x b + r ไม่ลงตัว
- นั่นคือเราพิสูจน์ได้ว่า g หาร b และ r ลงตัว (ยังไม่จบ)

- สมมติว่าเรามี g’ = GCD(b, r) เนื่องจาก g’ หารทั้ง b และ r ลงตัวดังนั้นมันจะหาร k x b + r ลงตัวเช่นกัน ดังนั้น g’ หาร a ลงตัว
- ต่อไป g และ g’ เป็นสองจำนวนที่แตกต่างกันใช่ไหม
- เราจะพิสูจน์ด้วย **contradiction** (เราก็ให้มันต่างกัน ก็ต้องมีตัวหนึ่งมากกว่าอีกตัว) สมมติให้ g > g’ เรารู้ว่า g หาร b และ r ลงตัว ดังนั้น GCD(b, r) จะเป็น g’ ได้อย่างไร(เป็นค่ามากสุดที่หารได้) เมื่อเรามีจำนวนที่มากกว่า g’ ที่หาก b และ r ได้ลงตัว
- ดังนั้น g ไม่มีทางมากกว่า g’
- ใช้วิธีเดียวกันในการบอกว่าว่า g’>g ดังนั้นทางเดียวที่จะเป็นไปได้คือ g = g’ ทำให้ได้ว่า GCD(a, b) = GCD(b, r) = GCD(b, a%b)

คุณสมบัติของ GCD

- GCD(a, b) = GCD(b, a)
- GCD ของ สามจำนวนสามารถคำนวณได้ด้วย

  GCD(a, b, c) = GCD(GCD(a, b), c)
- GCD(a, b) = GCD(a–b, b)
- ข้อควรระวัง ตัว algorithm นี้ทำงานได้ถูกต้องสำหรับข้อมูลเข้าที่เป็น non-negative เท่านั้น ลอง GCD(4,–2) คำตอบที่ถูกต้องเป็น 2 วิธีแก้ก็คือ ให้ส่งข้อมูลเข้าเป็นด้วยค่า absolute หรือ ใช้ค่า absolute ของคำตอบก็ได้
- อีกอย่าง GCD(0,0) =0 มันควรตอบ infinity