

ใส่ข้อมูลใหม่ขนาด 4 บิตเข้าไป → ได้ 12 บิตใหม่ = 6 บิต Runtime

Wpawee direct ของเขา

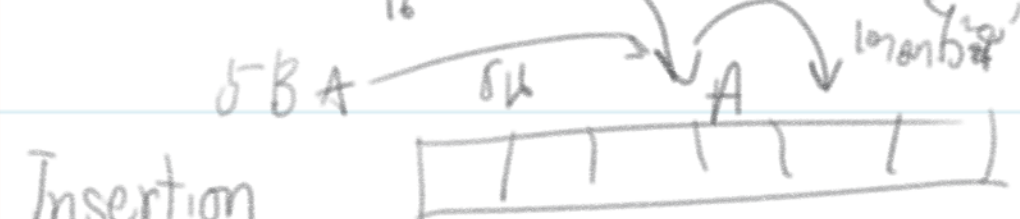
Open Addressing

$\alpha < 1$ เสมอ

Linear Probing ← ค้นหาต่อไปเรื่อยๆ

ใส่ข้อมูลสุดท้ายขนาด 16

ที่ว่าง → 6B 7C → 12

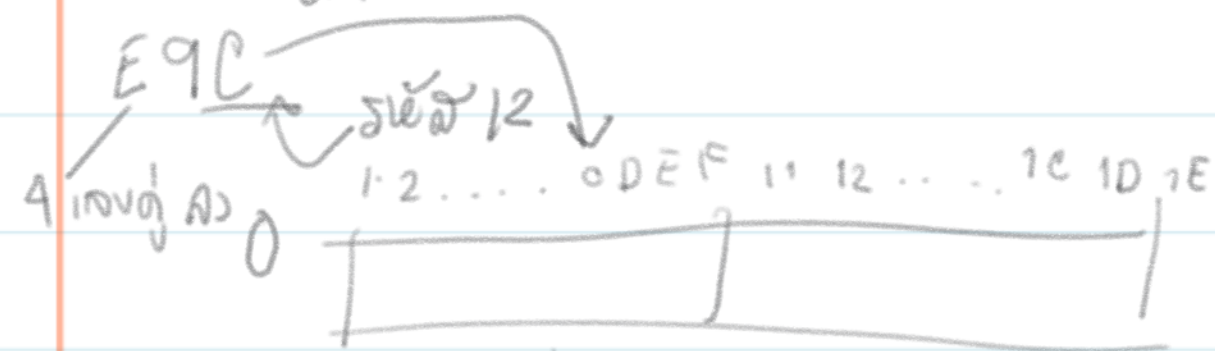


Insertion

ตรวจสอบอีกที

Resizing the array

least-significant 5 bit



Searching → ถ้าไม่เจอ → ค้นหาต่อไปเรื่อยๆ

Erasing → ลบข้อมูลไม่ได้ → ค้นหาต่อไปเรื่อยๆ → ถ้าเจอ → ลบออก → ถ้าไม่เจอ → ค้นหาต่อไปเรื่อยๆ

Primary Clustering



$$\text{อัตราส่วน} = \frac{k+2}{n}$$

การลบข้อมูลไม่ได้อันไหน

$$\text{ในกรณีที่ } \frac{1}{2} \left[1 + \frac{1}{1-\lambda} \right] \leftarrow \text{ถ้าเจอแล้ว}$$

$$\text{หาไม่เจอแล้ว } \frac{1}{2} \left(1 + \frac{1}{(1-\lambda)^2} \right) \leftarrow \text{ถ้าเจอแล้ว}$$

Run-time analysis ที่น่าสนใจ

ตรวจสอบ = ?

มี 3 มด

→ กำหนด M จำนวน

→ ใช้ Linear Probing

→ ใช้ Quadratic probing, hash

Quadratic Probing

Linear → ค้นหาต่อไปเรื่อยๆ

Quadratic → ค้นหาต่อไปเรื่อยๆ (ข้อที่ 1 + K²) / m

→ ถ้าเจอที่ใดก็หยุดที่ T²

กำหนดขนาดของตารางเป็นจำนวนเฉพาะ → มี p เล็กๆ

→ ถ้า rehash (x,2) ที่ตัวจำนวนเฉพาะเปลี่ยนไป → จำนวนน้อย

$$\text{ลองหาค่าใหม่ } \rightarrow \frac{k^2 + k}{2} \text{ เมื่อ } M = 2^m \leftarrow \text{จำนวนเต็ม}$$

มีตัวอย่าง (5)

ถ้าเจอที่ใดก็หยุดที่ T² ตรวจสอบ!!

int bin = hash_M(x, M);

for (int K=0; K<M; ++K)

bin = (bin+K)%M;

resize ← x2, เปลี่ยนเป็น 5 bits

รู้ไว้ไว้

Deletion → ลบข้อมูลออก

ใช้ Lazy deletion → ยังไม่ลบ แต่ติดค่า = 0 → รอจนกว่าจะลบออกด้วยลบ

ใช้ hash แบบ quadratic ไปทำไม? → ใช้ cache ที่ถูก