บทที่ 4

# การวิเคราะห์หาขนาดอุตตมะของบล็อคในแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี

ตามที่ได้กล่าวมาแล้วว่าขนาดของบล็อคของแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีมีความสำคัญต่อการประมวลผลอย่างมีประสิทธิภาพ โดยทั่ว ๆ ไปขนาดของบล็อคที่เหมาะสมจะขึ้นอยู่กับจำนวนระเบียนทั้งหมดภายในแฟ้มข้อมูล ขนาดความยาวของระเบียน ขนาดความยาวของคีย์ เทคนิคการค้นหา คีย์ในบล็อคดัชนี เทคนิคการค้นหาระเบียนในบล็อคระเบียน ตลอดจนขนาดของหน่วยความจำของระบบคอมพิวเตอร์ที่จะใช้ได้ และความเร็วของจานแม่เหล็กซึ่งใช้เป็นหน่วยช่วยความจำ จึงน่าจะมีมาตรการในการกำหนดขนาดของบล็อคสำหรับแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีอย่างเหมาะสม

การวิเคราะห์ในขั้นแรกจะพิจารณาหามาตรการในการกำหนดขนาดของบล็อคของแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี เพื่อให้เวลาที่ใช้ไปในการค้นหาระเบียนข้อมูลน้อยที่สุด โดยแยกพิจารณากรณีที่แฟ้มข้อมูลมีดัชนีเพียงระดับเดียวและกรณีที่มีดัชนีหลายระดับ การที่แฟ้มข้อมูลจะมีดัชนีระดับเดียวหรือหลายระดับนั้น ขนาดของหน่วยความจำของคอมพิวเตอร์จะเป็นตัวกำหนดที่สำคัญ กล่าวคือถ้าขนาดของบล็อคใหญ่มาก ข้อมูลในหน่วยความจำของคอมพิวเตอร์แต่ละครั้งจะมีจำนวนมากซึ่งอาจจะเกินเนื้อที่ที่มีอยู่สำหรับผู้ใช้ User available core) ในกรณีนี้ขนาดของบล็อคจึงจำเป็นต้องมีขนาดเล็กลงซึ่งอาจจะเป็นเหตุให้เกิดบล็อคดัชนีขึ้นมากกว่า 1 ระดับได้ ถ้าระเบียนทั้งหมดในแฟ้มข้อมูลมีจำนวนมาก

## 4.1 กรณีแฟ้มข้อมูลมีดัชนีเพียงระดับเดียว

ให้ N เป็นจำนวนระเบียนทั้งหมด

และ K เป็นจำนวนสูงสุดของระเบียนที่มีอยู่ได้ในบล็อคระเบียน

$\lceil N/K \rceil$ เป็นจำนวนเต็มที่น้อยที่สุดที่ไม่น้อยกว่า N/K

ดังนั้น $\lceil N/K \rceil$ จะเป็นจำนวนบล็อคระเบียนทั้งหมดที่มีอยู่ในแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี และเป็นจำนวนคีย์ ทั้งหมดภายในบล็อคดัชนีด้วยเช่นกัน

กำหนดให้ $L_r$ เป็นความยาวเชิงตรรก (Logical Length) ของระเบียนข้อมูล

และ $L_k$ เป็นความยาวเชิงตรรกของคีย์ จากโครงสร้างของข้อมูลภายในบล็อคระเบียนและบล็อคดัชนีดังรูปที่ 3.1 และรูปที่ 3.2 ตามลำดับทำให้ความยาวของบล็อคระเบียน จะต้องใช้เนื้อที่เท่ากับ $\left\lceil \dfrac{(L_r+L_k+2)+2}{112} \right\rceil 112$ คำ โดย Disk prep. factor เท่ากับ 112 และบล็อคดัชนีจะต้องมีขนาดไม่เล็กกว่า $(L_k+1)\lceil N/K \rceil+1$ คำ เพื่อจะได้เพียงพอสำหรับบรรจุ คีย์ทั้งหมดได้ในระดับเดียว ดังนั้นสภาวะต่อไปนี้จะต้องเป็นจริง

$$(L_k+1)\left\lceil N/K \right\rceil + 1 \leq \left\lceil \frac{(L_r+L_K+2)K+2}{112} \right\rceil 112 \quad (4.1)$$

และเนื้อที่ของหน่วยความจำของคอมพิวเตอร์ที่เหลืออยู่สำหรับผู้ใช้จะต้องไม่น้อยกว่า

$$\max\left((L_r+L_k+2)K,\ (L_k+1)\left\lceil N/K \right\rceil\right)$$

ในการค้นหาระเบียนที่ต้องการ จะต้องอ่านข้อมูลภายในบล็อคดัชนีทั้งหมดเข้าสู่หน่วยความจำของคอมพิวเตอร์ เพื่อทำการค้นหาที่อยู่ของบล็อคระเบียนซึ่งบรรจุระเบียนข้อมูลที่ต้องการค้นหา ลักษณะการค้นหาข้อมูลภายในบล็อคดัชนีเป็นแบบการค้นหาเชิงคู่ (Binary Search) จำนวนการเปรียบเทียบที่ใช้ในการค้นหาเชิงคู่ในกรณีนี้จะเท่ากับ $\left\lceil \log_2 \left\lceil N/K \right\rceil \right\rceil$ ดังนั้นเวลาที่ใช้ไปเพื่อค้นหาที่อยู่ของบล็อคระเบียนซึ่งระเบียนข้อมูลที่ต้องการบรรจุอยู่จะเท่ากับ

$$T_1 = T_a + T_t\,(L_k+1)\left\lceil N/K \right\rceil + T_c\left\lceil \log_2 \left\lceil N/K \right\rceil \right\rceil \quad (4.2)$$

โดย $T_a$ เป็นค่าเฉลี่ยของเวลาในการเคลื่อนย้ายหัวอ่านของจานแม่เหล็ก และความล่าช้าเนื่องจากการรอให้จานแม่เหล็กหมุนมายังตำแหน่งที่ต้องการจะอ่านข้อมูล (Average Access time + Rotational delay time)

$T_t$ เป็นเวลาที่ใช้ไปในการส่งข้อมูล 1 คำ จากจานแม่เหล็กซึ่งเป็นหน่วยช่วยความจำสู่หน่วยความจำของคอมพิวเตอร์

$T_c$ เป็นเวลาที่ใช้ไปในการเปรียบเทียบ 1 ครั้งซึ่งเท่ากับเวลาวัฏฏะ (Cycle Time)

เมื่อค้นหาที่อยู่ของบล็อคระเบียนที่บรรจุระเบียนที่ต้องการได้แล้ว จึงทำการอ่านและส่งข้อมูลจากบล็อคระเบียนนั้นจากหน่วยช่วยความจำสู่หน่วยความจำของคอมพิวเตอร์เพื่อทำการค้นหาระเบียนข้อมูลที่ต้องการ ซึ่งลักษณะการค้นหา จะใช้เทคนิคการค้นหาแบบเชิงคู่สำหรับบล็อคระเบียนที่ยังไม่เคยถูกทำให้เป็นปัจจุบัน และจะใช้เทคนิคการค้นหาแบบอนุบรรพสำหรับบล็อคระเบียนที่ได้มีการปรับให้เป็นปัจจุบันแล้ว ในที่นี้จะพิจารณาเฉพาะกรณีที่บล็อคระเบียนยังไม่มีการปรับให้เป็นปัจจุบันเลย เพื่อเป็นมาตรการกำหนดขนาดอุตตมะของบล็อค ดังนั้นเวลาที่ใช้ไปในการค้นหาระเบียนที่ต้องการภายในบล็อคระเบียนอาจเขียนได้ดังนี้

$$T_2 = T_a + T_t(L_r+L_k+2)K + T_c\left\lceil \log_2 k \right\rceil \qquad (4.3)$$

ดังนั้นจะเห็นว่า เวลาทั้งหมดที่จะต้องใช้ไปเพื่อการค้นหาระเบียนที่ต้องการในกรณีที่แฟ้มข้อมูลมีดัชนีเพียงระดับเดียว และใช้เนื้อที่ในหน่วยความจำของคอมพิวเตอร์เพียง 1 บัฟเฟอร์ (buffer) จะเป็นดังนี้

$$T = a + T_t\left((L_k+1)\left\lceil \frac{N}{K} \right\rceil + (L_r+L_k+2)K\right) + T_c\left(\left\lceil \log_2\left\lceil \frac{N}{K} \right\rceil \right\rceil + \left\lceil \log_2 K \right\rceil\right) \qquad (4.4)$$

โดยที่ a เป็นค่าคงที่ซึ่งไม่ขึ้นกับจำนวนระเบียนและขนาดของบล็อค

ค่าคงที่ a จะสูงกว่า $2T_a$ เล็กน้อย ทั้งนี้เพราะรวมโอเวอร์เฮด (Overhead) ของ CPU ด้วย

เมื่อพิจารณาสมการ (4.4) ให้ถ่องแท้จะเห็นว่า ถ้า K มีค่าเพิ่มขึ้น ค่าของ $T_1$ ในสมการ (4.2) จะมีค่าไม่สูงขึ้น (Nonincreasing) แต่ตรงกันข้ามจะมีผลให้ $T_2$ ในสมการ (4.3) สูงขึ้น และถ้า K มีค่าลดลง $T_1$ ในสมการ (4.2) จะมีค่าไม่ลดลง (Nondecreasing) แต่จะมีผลให้ $T_2$ ในสมการ (4.3) ลดลง ดังนั้น จึงอาจกล่าวได้ว่า $T_1$ เป็นฟังก์ชันแบบไม่เพิ่ม (Non-Increasing Function) ของ K และ $T_2$ เป็นฟังก์ชันแบบเพิ่ม (Increasing function) ของ K

จึงทำให้ T ในสมการ (4.4) มีค่าต่ำสุดเพียงจุดเดียว (unique minimum) ทั้งนี้เพราะ T ในสมการ (4.4) มีลักษณะเป็นฟังก์ชันเว้า (convex function)

ถ้ากำหนดให้ $K^*$ เป็นค่าอุตตมะของจำนวนสูงสุดของระเบียนภายในบล็อคระเบียนที่ทำให้เวลาในการค้นหามีค่าต่ำสุด จากคำนิยามของค่าต่ำสุดอาจเขียนเป็นอสมการได้ดังนี้

$$T_t\left((L_k+1)\left(\left\lceil \frac{N}{K^*}\right\rceil - \left\lceil \frac{N}{K^*-1}\right\rceil\right) + (L_r+L_k+2)\right)$$
$$+ T_c\left(\left\lceil \log_2 \left\lceil \frac{N}{K^*}\right\rceil\right\rceil - \left\lceil \log_2 \left\lceil \frac{N}{K^*-1}\right\rceil\right\rceil + \left\lceil \log_2 K^*\right\rceil - \left\lceil \log_2 (K^*-1)\right\rceil\right) \leq 0 \quad (4.5)$$

และ

$$T_t\left((L_k+1)\left(\left\lceil \frac{N}{K^*}\right\rceil - \left\lceil \frac{N}{K^*+1}\right\rceil\right) - (L_r+L_k+2)\right)$$
$$+ T_c\left(\left\lceil \log_2 \left\lceil \frac{N}{K^*}\right\rceil\right\rceil - \left\lceil \log_2 \left\lceil \frac{N}{K^*+1}\right\rceil\right\rceil + \left\lceil \log_2 K^*\right\rceil - \left\lceil \log_2 (K+1)\right\rceil\right) \leq 0 \quad (4.6)$$

ในกรณีที่ $K^*-1$ หรือ $K^*$ ไม่อาจเขียนให้อยู่ในรูป $2^m$ โดยที่ m เป็นค่าจำนวนเต็มบวกที่มากกว่า 1 จะทำให้

ความแตกต่างระหว่าง $\lceil \log_2 K^*\rceil$ กับ $\lceil \log_2 (K^*-1)\rceil$

และระหว่าง $\lceil \log_2 K^*\rceil$ กับ $\lceil \log_2 (K^*+1)\rceil$ จะมีค่าเท่ากับศูนย์ เช่น

$K^* = 35$

$$\lceil \log_2 K^*\rceil = \lceil \log_2 (K^*-1)\rceil = \lceil \log_2 (K^*+1)\rceil = 6$$

$K = 218$

$$\lceil \log_2 K^*\rceil = \lceil \log_2 (K^*-1)\rceil = \lceil \log_2 (K^*+1)\rceil = 8$$

ในทำนองเดียวกัน ถ้า $\left\lceil \frac{N}{K^*+1} \right\rceil$ หรือ $\left\lceil \frac{N}{K^*} \right\rceil$ ไม่อาจเขียนในรูป $2^m$ ได้

ความแตกต่างระหว่าง $\left\lceil \log_2 \left\lceil \frac{N}{K^*} \right\rceil \right\rceil$ กับ $\left\lceil \log_2 \left\lceil \frac{N}{K^*+1} \right\rceil \right\rceil$

และระหว่าง $\left\lceil \log_2 \left\lceil \frac{N}{K^*} \right\rceil \right\rceil$ กับ $\left\lceil \log_2 \left\lceil \frac{N}{K^*-1} \right\rceil \right\rceil$

จะมีค่าเท่ากับศูนย์

เมื่อเป็นเช่นนี้ ข้อสรุปจากอสมการ (4.5) และ (4.6) ในกรณีที่ $(K^*-1), K^*$

และ $\left\lceil \frac{N}{K^*} \right\rceil$ ไม่อาจเขียนให้อยู่ในรูป $2^m$ จะเป็นดังนี้

$$\left\lceil \frac{N}{K^*} \right\rceil - \left\lceil \frac{N}{K^*+1} \right\rceil \leq \frac{L_r+L_k+2}{L_k+1} \leq \left\lceil \frac{N}{K^*-1} \right\rceil - \left\lceil \frac{N}{K^*} \right\rceil \qquad (4.7)$$

เมื่อกำหนดค่า $N$, $L_k$ และ $L_r$ แล้วค่า $K$ ที่เหมาะสมจะต้องเป็นค่าของจำนวนบล็อคระเบียนทั้งหมดซึ่งสามารถบรรจุ ระเบียนข้อมูลทั้งแฟ้มโดยมีดัชนีระดับเดียวและเมื่อแทนค่า $K$ ในอสมการ (4.7) แล้วจะต้องเป็นจริงตามอสมการนั้นด้วย ถ้าหากไม่มีการประมาณค่าของ $K$ ก่อน ภาระการการคำนวณเพื่อหาค่าอุตตมะของ $K$ อาจจะมาก จึงน่าจะวิเคราะห์เพื่อประมาณค่าอุตตมะของ $K$ ดู การประมาณค่าอุตตมะของ $K$ กระทำได้โดยพิจารณา $K$ เป็นตัวแปรต่อเนื่อง (continuous variable) และมิต้องคำนึงถึง ค่าจำนวนเต็มของ $\left\lceil \frac{N}{K} \right\rceil$, $\log \left\lceil \frac{N}{K} \right\rceil$ และ $\left\lceil \log_2 K \right\rceil$

เมื่อเป็นเช่นนี้ สมการ (4.4) จะกลายเป็น

$$T = a+T_t\left((L_k+1)\frac{N}{K} + (L_r+L_k+2)K\right)+T_c\left(\log_2 \frac{N}{K} + \log_2 K\right) \qquad (4.8)$$

และอนุพันธ์ของ $T$ เมื่อเทียบกับ $K$ จะเขียนได้เป็น

$$\frac{dT}{dk} = T_t\left(-(L_k+1)\frac{N}{K^2} + L_r+L_k+2\right)$$

ซึ่งจะให้ค่าอุตตมะของ K โดยประมาณเป็น

$$K^{*} \simeq \sqrt{\frac{L_k + 1}{L_r + L_k + 2} \cdot N} \qquad (4.9)$$

$K^{*}$ ที่ได้นี้จะใช้เป็นค่าเริ่มต้นในการค้นหาค่าอุตตมะที่แท้จริงตามสมการ (4.7) อย่างไรก็ตามเมื่อได้ค่าอุตตมะแล้ว สภาวะตามอสมการ (4.1) จะต้องเป็นจริงด้วยเพื่อที่จะมีเนื้อที่บรรจุข้อมูลทั้งหมดโดยมีดัชนีเพียงระดับเดียวมิฉะนั้นจะต้องเพิ่มค่า K ขึ้นไปเพื่อเพิ่มเนื้อที่บรรจุข้อมูล การกระทำเช่นนี้มิได้ทำให้สภาพอุตตมะเสียไป ทั้งนี้เป็นเพราะ T ในอสมการ (4.4) มีลักษณะเป็นฟังก์ชันเว้า

การคำนวณหาค่าอุตตมะ K ในกรณีที่ยอมให้มีดัชนีเพียงระดับเดียว

<u>ตัวอย่างที่ 1</u> แฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีมีขนาด 249989 ระเบียน ระเบียนข้อมูลมีขนาดระเบียนละ 16 คำ และความยาวของคีย์ขนาด 3 คำ

$$K^{*} \simeq \sqrt{\frac{L_k + 1}{L_r + L_k + 2} \cdot N}$$

$L_k$ = 3

$L_r$ = 16

N = 249989

K* = 218.213

ลองนำค่า K* = 218 แทนในสมการ (4.7)

$$\left\lceil \frac{N}{K^*} \right\rceil - \left\lceil \frac{N}{K^*+1} \right\rceil \le \frac{L_r + L_k + 2}{L_k + 1} \le \left\lceil \frac{N}{K^*-1} \right\rceil - \left\lceil \frac{N}{K^*} \right\rceil$$

$5 \le 5.25 \le 6$ ซึ่งเป็นจริง

และเมื่อนำค่าอุตตมะ $K^*$ นี้แทนในอสมการ (4.1) ก็เป็นจริงเช่นกัน

$$(L_k+1)\lceil N/K\rceil \leq \left\lceil\frac{(L_r+L_k+2)K}{112}\right\rceil 112$$

$$4588 \leq 4592$$

ดังนั้นค่าอุตตมะ $K^*$ ที่เหมาะสมสำหรับแฟ้มข้อมูลดังกล่าวคือ 218 ระเบียน

ในกรณีที่บล็อคระเบียนได้เคยถูกปรับให้เป็นปัจจุบันแล้ว การค้นหาข้อมูลที่ต้องการภายในบล็อคระเบียนจะใช้เทคนิคการค้นหาแบบอนุบรรพ ซึ่งจำนวนการเปรียบเทียบในการค้นหาระเบียนที่ต้องการเท่ากับ $\lceil K/2\rceil$ แต่การค้นหาตำแหน่งของบล็อคระเบียนภายในบล็อคดัชนียังคงเป็นแบบการค้นหาเชิงคู่ เพราะว่าระเบียนต่าง ๆ ภายในบล็อคดัชนียังคงเหมือนเดิมไม่มีการเปลี่ยนแปลงตลอดกาล ดังนั้นเวลาที่ใช้ไปในการหาตำแหน่งบล็อคระเบียนจากบล็อคดัชนียังคงเป็นไปตาม $T_1$ ในสมการ (4.2) แต่เวลาการค้นหาภายในบล็อคระเบียนจะกลายเป็น

$$T_3 = T_a+T_t(L_r+L_k+2)+T_c\left\lceil\frac{K}{2}\right\rceil \qquad (4.10)$$

จึงทำให้เวลาค้นหาทั้งหมดกลายเป็น

$$T_s = a+T_t\left((L_k+1)\left\lceil\frac{N}{K}\right\rceil+(L_r+L_k+2)K+T_c\left(\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil+\left\lceil\frac{K}{2}\right\rceil\right)\right) \qquad (4.11)$$

โดยที่ $a$ เป็นค่าคงที่ซึ่งไม่ขึ้นกับจำนวนระเบียนและขนาดของบล็อค

ถ้า $K$ มีค่าสูงขึ้นจะไม่ทำให้ $T_1$ สูงขึ้น แต่จะทำให้ $T_3$ ในสมการ (4.10) สูงขึ้น และถ้า $K$ มีค่าลดลงจะไม่ทำให้ $T_1$ มีค่าลดลง แต่จะทำให้ $T_3$ มีค่าลดลง ดังนั้นจะทำให้ค่า $T$ ในสมการ (4.11) มีค่าต่ำสุดเพียงจุดเดียวเช่นเดียวกับสมการ (4.4)

กำหนดให้ $K^*$ เป็นค่าอุตตมะของจำนวนสูงสุดของระเบียนภายในบล็อคระเบียนที่ทำให้เวลาในการค้นหามีค่าต่ำสุด จากสมการ (4.11) อาจเขียนเป็นอสมการได้ดังนี้

$$T_t\left((L_k+1)\left(\left\lceil\frac{N}{K^*}\right\rceil-\left\lceil\frac{N}{K^*-1}\right\rceil\right)+L_r+L_k+2\right)$$
$$T_c\left(\left\lceil\log_2\left\lceil\frac{N}{K^*}\right\rceil\right\rceil-\left\lceil\log_2\left\lceil\frac{N}{K^*-1}\right\rceil\right\rceil+\left\lceil\frac{K^*}{2}\right\rceil-\left\lceil\frac{K^*-1}{2}\right\rceil\right)\leq 0 \quad (4.12)$$

และ

$$T_t\left((L_k+1)\left(\left\lceil\frac{N}{K^*}\right\rceil-\left\lceil\frac{N}{K^*+1}\right\rceil\right)-\left(L_r+L_k+2\right)\right)$$
$$+T_c\left(\left\lceil\log_2\left\lceil\frac{N}{K^*}\right\rceil\right\rceil-\left\lceil\log_2\left\lceil\frac{N}{K^*+1}\right\rceil\right\rceil+\left\lceil\frac{K^*}{2}\right\rceil-\left\lceil\frac{K^*+1}{2}\right\rceil\right)\leq 0 \quad (4.13)$$

ในกรณีที่ $\left\lceil\frac{N}{K^*+1}\right\rceil$ หรือ $\left\lceil\frac{N}{K^*}\right\rceil$ ไม่อาจเขียนให้อยู่ในรูปของ $2^m$ ได้ โดยที่ $m$ เป็นค่าเต็มบวกที่มีค่ามากกว่า 1 ความแตกต่างระหว่าง $\left\lceil\log_2\left\lceil\frac{N}{K^*}\right\rceil\right\rceil$ กับ $\left\lceil\log_2\left\lceil\frac{N}{K^*+1}\right\rceil\right\rceil$ และระหว่าง $\left\lceil\log_2\left\lceil\frac{N}{K^*}\right\rceil\right\rceil$ กับ $\left\lceil\log_2\left\lceil\frac{N}{K^*-1}\right\rceil\right\rceil$ จะมีค่าเท่ากับศูนย์ ดังนั้นอสมการ (4.13) และ (4.14) จะกลายเป็น

$$\frac{L_r+L_k+2}{L_k+1}+\frac{T_c}{T_t(L_k+1)}\left(\left\lceil\frac{K^*}{2}\right\rceil-\left\lceil\frac{K^*-1}{2}\right\rceil\right)\leq\left\lceil\frac{N}{K^*-1}\right\rceil-\left\lceil\frac{N}{K^*}\right\rceil \quad (4.14)$$

และ

$$\frac{L_r+L_k+2}{L_k+1}+\frac{T_c}{T_t(L_k+1)}\left(\left\lceil\frac{K^*+1}{2}\right\rceil-\left\lceil\frac{K^*}{2}\right\rceil\right)\geq\left\lceil\frac{N}{K^*}\right\rceil-\left\lceil\frac{N}{K^*+1}\right\rceil \quad (4.15)$$

ส่วนผลต่างระหว่าง $\left\lceil\frac{K^*}{2}\right\rceil$ กับ $\left\lceil\frac{K^*-1}{2}\right\rceil$ จะมีค่าเท่ากับศูนย์ ถ้า $K^*$ เป็นเลขคู่ และจะมีค่าเท่ากับ 1 ถ้า $K^*$ เป็นเลขคี่ และผลต่างระหว่าง $\left\lceil\frac{K^*+1}{2}\right\rceil$ กับ $\left\lceil\frac{K^*}{2}\right\rceil$

จะมีค่าเท่ากับ 1 ถ้า K เป็นเลขคู่ และเท่ากับศูนย์ ถ้า $K^*$ เป็นเลขคี่ จึงอาจกล่าวได้ว่า

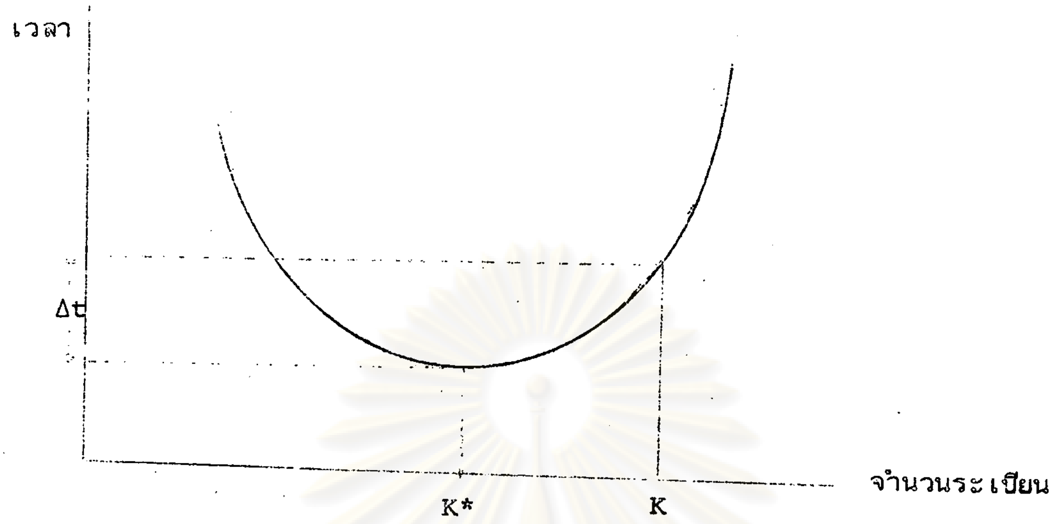
$$\frac{T_c}{T_t(L_k+1)}\left(\left\lceil \frac{K^*}{2} \right\rceil - \left\lceil \frac{K^*-1}{2} \right\rceil\right) \leq \frac{L_r+L_k+2}{L_k+1} \qquad (4.16)$$

ดังนั้นอสมการ (4.14) และ (4.15) จึงสรุปได้เป็น

$$\left\lceil \frac{N}{K^*} \right\rceil - \left\lceil \frac{N}{K^*+1} \right\rceil \leq \frac{L_r+L_k+2}{L_k+1} \leq \left\lceil \frac{N}{K^*-1} \right\rceil - \left\lceil \frac{N}{K^*} \right\rceil \qquad (4.17)$$

ซึ่งเหมือนกับอสมการ (4.7) ในกรณีที่การค้นหาข้อมูลภายในบล็อคระเบียนเป็นแบบการค้นหาเชิงคู่

การวิเคราะห์ดังกล่าวในการหาค่าอุตตมะของจำนวนระเบียนภายในบล็อคระเบียนของแฟ้มข้อมูลทั้งกรณีที่แฟ้มข้อมูลยังไม่เคยถูกปรับให้เป็นปัจจุบันและกรณีที่เคยถูกปรับให้เป็นปัจจุบันแล้ว อาจกล่าวได้ว่าขนาดอุตตมะของจำนวนระเบียนภายในบล็อคระเบียนสำหรับแฟ้มข้อมูลทั้งสองกรณีจะไม่ต่างกันมากนัก จึงทำให้การกำหนดขนาดของอุตตมะของบล็อคเป็นไปได้ง่ายขึ้น และสามารถประยุกต์ใช้กับแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี ซึ่งในทางปฏิบัติบล็อคระเบียนจะมีทั้งที่ถูกปรับให้เป็นปัจจุบันแล้วและที่ยังไม่ได้ถูกปรับให้เป็นปัจจุบันปะปนกันอยู่ อย่างไรก็ตามการออกแบบแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีให้มีบล็อคดัชนีเพียงระดับเดียวก็มิได้หมายความว่าจะเหมาะสมเสมอไปทั้งนี้เพราะในบางกรณีขนาดอุตตมะของบล็อคระเบียนที่เป็นจริงตามอสมการ (4.7) แต่ไม่เป็นจริงตามอสมการ (4.1) จึงทำให้ต้องเพิ่มขนาดของบล็อคให้โตขึ้นเพื่อให้มีเนื้อที่เพียงพอที่จะบรรจุข้อมูลทั้งหมดได้โดยมีดัชนีเพียงระดับเดียว




ดังนั้น ถ้าเมื่อใดที่ผลต่างของเวลาที่ใช้ไปสำหรับแฟ้มข้อมูลที่มีขนาดบล็อค $K^*$ กับขนาดบล็อค K มีค่ามากกว่าเวลาที่ใช้ไปในการหาตำแหน่งบล็อคบนจานแม่เหล็ก (Access time) จึงสมควรที่จะยอมให้มีดัชนีถึง 2 ระดับ ทั้งนี้เพราะการมีดัชนีระดับเดียว ถึงแม้ว่าจะใช้เวลาในการค้นหาบล็อคดัชนีเพียงครั้งเดียวก็จริงแต่เนื่องจากจำนวนระเบียนภายในบล็อคระเบียนมีมากทำให้ต้องใช้เวลาไปในการขนย้ายข้อมูลทั้งบล็อคนั้นเข้าสู่หน่วยความจำของคอมพิวเตอร์เพื่อค้นหาข้อมูลที่ต้องการเพียงระเบียนเดียวกันและเป็นที่ทราบกันดีว่า การที่บล็อคระเบียนมีระเบียนจำนวนมากจะทำให้ต้องใช้เวลาในการค้นหาระเบียนที่ต้องการภายในบล็อคระเบียนนานกว่าบล็อคที่มีระเบียนจำนวนน้อย แต่การค้นหาตำแหน่งของบล็อคระเบียนภายในบล็อคดัชนีจะเร็วกว่า เพราะบล็อคระเบียนที่บรรจุระเบียนได้มากย่อมมีจำนวนบล็อคน้อยกว่าบล็อคระเบียนที่บรรจุระเบียนจำนวนน้อย ดังนั้นถ้าขนาดของบล็อค K ต่างไปจาก $K^*$ มากจนทำให้ผลต่างของเวลาเกินกว่าเวลาที่ใช้ไปในการหาตำแหน่งบล็อคดัชนีหรือบล็อคระเบียน ($T_a$) จึงสมควรที่จะยอมให้มีดัชนี 2 ระดับ

## 4.2 กรณีแฟ้มข้อมูลมีบล็อคดัชนีสองระดับ

ดังที่ได้กล่าวมาแล้วว่า เนื้อที่สำหรับหน่วยความจำของคอมพิวเตอร์มีอยู่จำนวนจำกัดและถ้าระบบคอมพิวเตอร์มีเนื้อที่ดังกล่าวเหลือไว้ให้ผู้ใช้เพียงเล็กน้อยจนไม่สามารถกำหนดขนาดบล็อคของแฟ้มข้อมูลให้โตพอสำหรับที่จะมีดัชนีเพียงระดับเดียวได้ จึงจำเป็นต้องยอมให้แฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีมีดัชนีถึง 2 ระดับ ในการวิเคราะห์เพื่อหาค่าอุตตมะของจำนวนสูงสุดของระเบียนภายในบล็อคระเบียนสามารถกระทำได้ดังนี้

กำหนดให้ $K_1$ และ $K_2$ เป็นค่าอุตตมะของจำนวนสูงสุดของระเบียนภายในบล็อคดัชนีระดับที่หนึ่ง และระดับที่สองตามลำดับที่ทำให้เวลาในการค้นหามีค่าต่ำสุด

$N$ เป็นจำนวนระเบียนของข้อมูลทั้งหมดภายในแฟ้มข้อมูล ดังนั้น เวลาในการค้นหาบล็อคดัชนีระดับที่หนึ่งและระดับที่สอง เขียนได้ดังนี้

$$T_1 = T_a+T_t(L_k+1)K_1+T_c\left\lceil\log_2 K_1\right\rceil \tag{4.18}$$

$$T_2 = T_a+T_t(L_k+1)K_2+T_c\left\lceil\log_2 K_2\right\rceil \tag{4.19}$$

และเวลาที่ใช้ไปในการค้นหาบล็อคระเบียนเขียนได้เป็น

$$T_3 = T_a+T_t(L_r+L_k+2)\left\lceil\frac{N}{K_1K_2}\right\rceil+T_c\left\lceil\log_2\left\lceil\frac{N}{K_1K_2}\right\rceil\right\rceil \tag{4.20}$$

∴ เวลาทั้งหมดที่ใช้ไปในการค้นหาระเบียนที่ต้องการในกรณีที่แฟ้มข้อมูลมีดัชนีถึงสองระดับคือ

$$T = a+T_t\left((L_k+1)(K_1+K_2)+(L_r+L_k+2)\left\lceil\frac{N}{K_1K_2}\right\rceil\right)+T_c\left(\left\lceil\log_2 K_1\right\rceil+\left\lceil\log_2 K_2\right\rceil+\left\lceil\log_2\left\lceil\frac{N}{K_1K_2}\right\rceil\right\rceil\right) \tag{4.21}$$

โดยที่ $a$ เป็นค่าซึ่งไม่ขึ้นกับจำนวนระเบียนและขนาดของบล็อคระเบียน พิจารณาสมการ (4.21) จะเห็นว่า ถ้า $K_1$ และ $K_2$ มีค่าเพิ่มขึ้น จะทำให้ค่า $T_1$ และ $T_2$ ในสมการ (4.18) และ (4.19) สูงขึ้น แต่ไม่ทำให้ $T_3$ ในสมการ (4.20) สูงขึ้น ดังนั้นจึงอาจกล่าวได้ว่า $T_3$ เป็นฟังก์ชันที่ไม่เพิ่มของ $K$ ในขณะที่ $T_1$ และ $T_2$ เป็นฟังก์ชันที่เพิ่มของ $K$

ในกรณีตัวแปร 2 ตัว การวิเคราะห์เพื่อให้ได้สมการ ทำนองเดียวกันกับสมการ (4.7) มิอาจกระทำได้ง่าย จึงจะขอทำการวิเคราะห์อย่างประมาณโดยให้ $K_1$ และ $K_2$ เป็นตัวแปรต่อเนื่องตลอดจนมิได้สนใจว่า จำนวนครั้งสูงสุดในการเปรียบเทียบจะเป็นเลขจำนวนเต็มหรือไม่ ดังนั้นสมการ (4.21) จะกลายเป็น

$$T = a+T_t\left((L_k+1)(K_1+K_2)+(L_r+L_k+2)\frac{N}{K_1K_2}\right)+T_d\left(\log_2K_1+\log_2K_2+\log_2\frac{N}{K_1K_2}\right) \quad (4.22)$$

อนุพันธ์ของ T เมื่อเทียบกับ $K_1$ และ $K_2$ จะเขียนได้ดังสมการ (4.23)และ (4.24)

$$\frac{\partial T}{\partial K_1} = T_t\left((L_k+1)+(L_r+L_k+2)\left(\frac{-N}{K_2K_1^2}\right)\right) \quad (4.23)$$

และ

$$\frac{\partial T}{\partial K_2} = T_t\left((L_k+1)+(L_r+L_k+2)\left(\frac{-N}{K_1K_2^2}\right)\right) \quad (4.24)$$

จากสมการ (4.23) และ (4.24) จะได้ว่า

$K_2K_1^2$ และ $K_1K_2^2$ มีค่าเท่ากันคือ $\dfrac{L_r+L_k+2}{L_k+1}\cdot N$

ดังนั้นจึงสรุปได้ว่าค่าอุตตมะโดยประมาณของ $K_1$ และ $K_2$ มีค่าเท่ากัน กำหนดให้ $K^*$ แทนค่าอุตตมะของ $K_1$ และ $K_2$ สมการ (4.23) หรือ (4.24) จะเขียนได้เป็นดังนี้

$$K^* \simeq \sqrt[3]{\frac{L_r+L_k+2}{L_k+1}\cdot N} \quad (4.25)$$

จากการที่ยอมให้แฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีมีดัชนีถึง 2 ระดับ ทำให้ทุกครั้งที่มีการค้นหาระเบียนข้อมูลที่ต้องการจะต้องใช้เวลาไปในการหาบล็อคดัชนีเพิ่มขึ้นอีก 1 ครั้งและเป็นที่ทราบดีแล้วว่าการหาตำแหน่งบล็อคบนจานแม่เหล็กนั้นใช้เวลานาน ซึ่งน่าที่จะหาเทคนิคอื่นเข้ามาช่วยเพื่อหลีกเลี่ยงการใช้เวลาที่เพิ่มขึ้นดังกล่าว เทคนิคหนึ่งซึ่งนิยมนำมาใช้คือ การแบ่งแฟ้มข้อมูลหลักออกเป็นแฟ้มข้อมูลย่อย แต่การใช้เทคนิคดังกล่าวนี้ย่อมจะมีโอเวอร์เฮด เกิดขึ้นบ้าง ซึ่งส่วนใหญ่เป็นเวลาที่ใช้ไปเกี่ยวกับ CC/ER ดังนั้นในการที่จะนำเทคนิคนี้มาใช้ควรพิจารณาดังนี้

กำหนดให้ $N$ เป็นจำนวนระเบียนทั้งหมดของข้อมูลภายในแฟ้ม

$K_1$ เป็นค่าอุตตมะของจำนวนระเบียนสูงสุดภายในบล็อคระเบียน เมื่อแฟ้มข้อมูลถูกออกแบบให้มีดัชนีเพียงระดับเดียว

$K_2$ เป็นค่าอุตตมะของจำนวนระเบียนสูงสุดภายในบล็อคระเบียน เมื่อแฟ้มข้อมูลถูกออกแบบให้มีดัชนี 2 ระดับ

$\Delta t$ คือผลต่างของเวลาที่ใช้ไปในการค้นหาระเบียนที่ต้องการหนึ่งระเบียนสำหรับแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีที่มีค่าอุตตมะของจำนวนระเบียนสูงสุดภายในบล็อคเป็น $K_1$ และ $K_2$

$H$ เป็นโอเวอร์เฮด ที่เพิ่มขึ้นเมื่อใช้แฟ้มข้อมูลย่อย

$M$ เป็นจำนวนทั้งหมดของข้อมูลเปลี่ยนแปลงโดยเฉลี่ยที่ถูกนำมาประมวลผลในแต่ละครั้ง

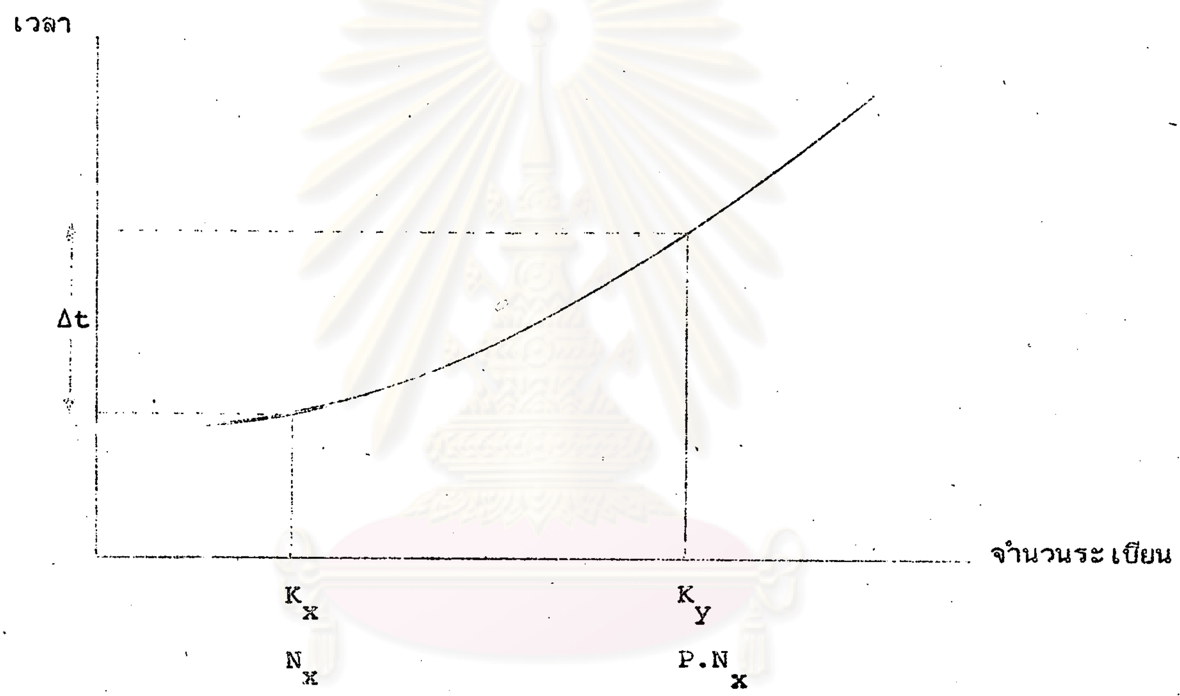
ดังนั้น เมื่อใดที่ $H < M\Delta t$ จึงสมควรที่จะนำเทคนิคดังกล่าวมาใช้

และในทำนองเดียวกัน มักจะพบเสมอว่าถ้าแฟ้มข้อมูลมีขนาดใหญ่ขึ้นถึงระดับหนึ่งแล้ว ประสิทธิภาพในการใช้แฟ้มข้อมูลจะลดลงอย่างมาก ถึงแม้ว่าแฟ้มข้อมูลจะได้ออกแบบอย่างถูกต้องและมีดัชนีเพียงระดับเดียวก็ตาม ดังนั้นควรจะพิจารณาดังนี้

กำหนดให้ $K_x$ เป็นค่าอุตตมะของจำนวนระเบียนสูงสุดภายในบล็อคระเบียนของแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีซึ่งมีดัชนีเพียงระดับเดียว และแฟ้มมีขนาด $N_x$ ระเบียน

$K_y$ เป็นค่าอุตตมะของจำนวนระเบียนสูงสุดเมื่อแฟ้มข้อมูลมีดัชนีเพียงระดับเดียวเช่นกันและแฟ้มข้อมูลมีขนาดเป็นทวีคูณของ $N_x$ เท่ากับ P เท่า

$\Delta t$ เป็นผลต่างของเวลาในการค้นหาระเบียนที่ต้องการ 1 ระเบียน ภายในแฟ้มข้อมูลขนาด $N_x$ และ $P.N_x$




H เป็นโอเวอร์เฮด ที่เกิดขึ้นเมื่อมีการใช้แฟ้มข้อมูลย่อย

M เป็นจำนวนระเบียนทั้งหมดของข้อมูลเปลี่ยนแปลงโดยเฉลี่ยที่จะนำมาประมวลผลในแต่ละครั้ง

ดังนั้น เมื่อใดที่ $H < M \Delta t$ จึงสมควรที่จะแบ่งแฟ้มข้อมูลออกเป็นแฟ้มข้อมูลย่อย

4.3 การกำหนดขนาดของบล็อคและการค้นหาระเบียนข้อมูลภายในแฟ้มข้อมูล แบบอนุบรรพเชิงดัชนีซึ่งจัดการโดย พีซีไอโอเอส (PCIOS) level 3R1A

ให้ N เป็นจำนวนระเบียนข้อมูลทั้งหมด และ K เป็นจำนวนสูงสุดของระเบียนที่มีอยู่ในแต่ละบล็อคระเบียน

$\left\lceil \frac{N}{K} \right\rceil$ เป็นจำนวนเต็มที่น้อยที่สุดแต่ไม่น้อยกว่า $\frac{N}{K}$

ดังนั้น $\left\lceil \frac{N}{K} \right\rceil$ จะเป็นจำนวนบล็อคระเบียนทั้งหมดที่มีอยู่ในแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนี และจะเป็นจำนวนคีย์ ทั้งหมดในบล็อคดัชนีสำหรับแฟ้มข้อมูลที่มีดัชนีเพียงระดับเดียว

$L_r$ เป็น ความยาวของระเบียนข้อมูล

$L_k$ เป็น ความยาวของ คีย์

เนื่องจากโครงสร้างของบล็อคดัชนีซึ่งจัดการโดย PCIOS มีลักษณะดังรูปที่ 3.1

ดังนั้น บล็อคดัชนีจะต้องมีขนาดเท่ากับ $\left\lceil \frac{(L_k+1)\left\lceil N/K \right\rceil + 1}{112} \right\rceil 112$

โดย 112 คือ Disk prep. factor.

ส่วนบล็อคระเบียนซึ่งมีโครงสร้างดังรูปที่ 3.2 นั้น พีซีไอโอเอสจะต้องใช้เนื้อที่นอกเหนือจากที่จะบรรจุระเบียนข้อมูลอีก 1 คีย์ และ 2 word address โดยถ้าเป็นบล็อคระเบียนแรก คีย์แรกสุดของบล็อคจะมีค่าศูนย์ (เลขฐานสอง) ซึ่งเป็นค่าต่ำสุดเพื่อไว้สำหรับการ insert ระเบียนที่มีค่าต่ำกว่าระเบียนแรกของแฟ้มข้อมูล สำหรับคีย์แรกของบล็อคระเบียนอื่น ๆ จะเป็นคีย์สุดท้ายของบล็อคระเบียนที่แล้ว

กำหนดให้ $B_s$ เป็นขนาดของบล็อคดังกล่าว

$\therefore$ จำนวนคีย์ ที่จะบรรจุได้ในบล็อคดัชนีจะเท่ากับ $\left\lfloor \frac{B_s-1}{L_k+1} \right\rfloor$

ในการค้นหาระเบียนข้อมูลภายในแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีที่ยังไม่เคยถูกทำให้เป็นปัจจุบันซึ่งการค้นหาเป็นแบบเชิงคู่ กรณีที่แฟ้มข้อมูลมีดัชนีเพียงระดับเดียว พีซีไอโอเอสจะต้องอ่านบล็อคดัชนีและส่งข้อมูลทั้งบล็อคเข้าสู่หน่วยความจำของคอมพิวเตอร์ เพื่อหาดูว่าตำแหน่งบล็อคระเบียนที่ต้องการอยู่ที่ใดแล้วจึงทำการอ่านและส่งข้อมูลบล็อคระเบียนนั้นเข้าสู่หน่วยความจำ เพื่อค้นหาระเบียนที่ต้องการ ดังนั้นเวลาที่ใช้ไปเพื่อการค้นหาระเบียนหนึ่ง ๆ เมื่อมี 1 บัฟเฟอร์จะเป็นดังนี้

$$T_i = a_1+2T_t\ B_s\ +T_c\left(\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil+\left\lceil\log_2 K\right\rceil\right) \qquad (4.26)$$

โดยที่ $a_1$ มีค่าสูงกว่า $T_a$ เล็กน้อย

และ $B_s$ แทนขนาดของบล็อคสำหรับแฟ้มข้อมูลที่มีดัชนีเพียงระดับเดียว

เมื่อพิจารณาสมการ (4.26) จะพบว่า เมื่อ K มีค่าเพิ่มขึ้น $t_c\left\lceil\log_2 K\right\rceil$ จะมีค่าไม่ลดลง (non-decreasing) แต่ $t_c\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil$ จะมีค่าไม่เพิ่ม (non-increasing) และเมื่อ K มีค่าลดลง $t_c\left\lceil\log_2 K\right\rceil$ จะมีค่าไม่เพิ่ม แต่ $t_c\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil$ จะมีค่าไม่ลดลง การเพิ่มขึ้นหรือลดลงของเทอมเหล่านี้จะเป็นขั้น ๆ (step) จึงมิอาจใช้คุณสมบัติของฟังก์ชันเว้าได้ แต่เมื่อพิจารณา

$\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil$ และ $\log_2 K$ จะพบว่ามีการเปลี่ยนแปลงช้ามากเมื่อ K มีการเปลี่ยนแปลงและการเพิ่มขึ้นของ $\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil$ **มักจะถูกหักล้างด้วยการลดลงของ** $\left\lceil\log_2 K\right\rceil$ แต่ $B_s$ เปลี่ยนแปลงเร็วกว่า $\left\lceil\log_2\left\lceil\frac{N}{K}\right\rceil\right\rceil$ หรือ $\left\lceil\log_2 K\right\rceil$ เมื่อ K มีค่าเปลี่ยนแปลง เนื่องจาก K มี $(L_r+L_k+2)$ เป็นตัวคูณจึงอาจกล่าวได้ว่า การเคลื่อนไหวของ T ในสมการ (4.26) ถูกกำหนดโดย $B_s$ ดังนั้นบล็อคระเบียนจะต้องมีขนาดเท่ากับ $\left\lceil\frac{(L_r+L_k+2)K+L_k+4}{112}\right\rceil 112$ และเนื่องจากพีซีไอโอเอสกำหนดให้ขนาดของบล็อคดัชนีเท่ากับขนาดของบล็อคระเบียน ในการคำนวณหาค่าอุดตมะของบล็อคจะต้องเป็นจริงตามสภาวะ (condition) นี้

$$\left\lceil\frac{(L_k+1)\left\lceil N/K\right\rceil + 1}{112}\right\rceil 112 = \left\lceil\frac{(L_r+L_k+2)K+L_k+4}{112}\right\rceil 112$$

และเนื้อที่ความจำของคอมพิวเตอร์จะต้องไม่น้อยกว่า

$$\left(\left\lceil \frac{(L_k+1)\ \lceil N/K\rceil\ +\ 1}{112}\right\rceil\ 112,\ \left\lceil \frac{(L_r+L_k+2)K\ +L_k+4}{112}\right\rceil\ 112\right)$$

ในกรณีที่แฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีมีดัชนี 2 ระดับ เวลาที่ใช้ไปเพื่อการค้นหาระเบียนหนึ่ง ๆ จะเป็นดังนี้

$$T_{ii} = a_2+3T_tB_{ss}+T_c\left(\lceil\log_2K_1\rceil + \lceil\log_2K_2\rceil + \left\lceil \log_2\left\lceil\frac{N}{K_1K_2}\right\rceil\right\rceil\right) \qquad \textbf{(4.27)}$$

โดยที่ $a_2$ เป็นค่าที่สูงกว่า 3 $T_a$ เล็กน้อย

และ $B_{ss}$ แทนขนาดของบล็อคสำหรับแฟ้มข้อมูลที่มีดัชนี 2 ระดับ

เนื่องจาก $B_{ss}$ จะมีค่าน้อยกว่า $B_s$ ถ้าจำนวนข้อมูลเท่ากัน

ดังนั้นเมื่อพิจารณา $T_i$ ในสมการ (4.26) เทียบกับ $T_{ii}$ ในสมการ (4,27) อาจจะเกิดกรณีที่ $T_i$ มีค่าสูงกว่า $T_{ii}$ ได้เมื่อ

$2T_t\ B_s\ -\ 3T_t\ B_{ss}\ >\ T_a$ หรืออาจเขียนใหม่ได้เป็น

$$2B_s\ -3B_{ss}\ >\ T_a/T_t \qquad (4.28)$$

แต่ถ้าในการประมวลผลมีบัฟเฟอร์มากกว่า 1 และคงบล็อคดัชนีระดับสูงสุดไว้ในหน่วยความจำของคอมพิวเตอร์ตลอดการประมวลผล

ดังนั้น $T_i$ จะกลายเป็น $T_a+T_tB_s+T_c\lceil\log_2K\rceil$

และ $T_{ii}$ จะกลายเป็น $2T_a + 2T_tB_{ss}+T_c\left(\lceil\log_2 K_2\rceil+\left\lceil\log_2\left\lceil\frac{N}{K_1K_2}\right\rceil\right\rceil\right)$

ดังนั้นโอกาส $T_i$ จะมีค่าน้อยกว่า $T_{ii}$ ย่อมมีมากเนื่องจาก

$$T_t B_s - 2T_t B_{ss} < T_a \quad \text{หรือ}$$

$$B_s - 2B_{ss} < T_a/T_t$$

นอกจากนี้โอกาสที่ระเบียนที่จะค้นหาต่อมาอาจจะปรากฏอยู่ในบล็อคระเบียน $B_s$ ที่แล้วย่อมมี ทำให้ไม่ต้องใช้เวลาไอ/โอ เพื่อหาตำแหน่งบล็อคและส่งข้อมูลของบล็อคระเบียนนั้นเข้าสู่หน่วยความจำของคอมพิวเตอร์อีก

## 4.4 หลักการเลือกวิธีการเข้าถึงข้อมูล (Access Mode)

การเข้าถึงข้อมูลสำหรับแฟ้มข้อมูลแบบอนุบรรพเชิงดัชนีสามารถกระทำได้ 3 วิธี คือ

1. วิธีการเข้าถึงแบบอนุบรรพ (Sequential Mode)
2. วิธีการเข้าถึงแบบสุ่ม (Random Mode)
3. วิธีการเข้าถึงแบบพลวัต (Dynamic Mode)

การที่จะเลือกวิธีการเข้าถึงข้อมูลแบบใดจึงจะเหมาะสมนั้นควรจะได้พิจารณาถึงลักษณะของข้อมูลที่ต้องการค้นหาว่ามีลักษณะอย่างไร กล่าวคือถ้าข้อมูลที่ต้องการค้นหามีลักษณะต่อเนื่องกันโดยตลอด ควรจะเลือกการประมวลผลแบบอนุบรรพ เช่น แฟ้มข้อมูลหลักที่ต้องการค้นหาประกอบด้วยระเบียนซึ่งมีคีย์ตั้งแต่ 0001 ถึง 9999 และข้อมูลที่ต้องการจะค้นหาประกอบด้วยคีย์ต่อเนื่องตั้งแต่ 1111 ถึง 2500 ในระหว่างคีย์ดังกล่าวอาจจะมีคีย์ที่ไม่ต่อเนื่องอยู่บ้าง แต่ไม่ควรห่างกันเกินกว่าบีเอฟของแฟ้มข้อมูลนั้น เพราะจะทำให้เวลาไอ/โอที่ใช้ไปในการอ่านและส่งข้อมูลของบล็อคระเบียนนั้นสูญเปล่าเนื่องจากไม่มีคีย์ที่ต้องการปรากฏอยู่เลย ในการค้นหาไม่จำเป็นต้องเริ่มตั้งแต่คีย์ 0001 ควรจะ Sub-Sequential ตั้งแต่คีย์ 1111 เป็นต้นไป

ในกรณีที่คีย์เรียงลำดับต่อเนื่องเป็นช่วง ๆ ควรจะเลือกการเข้าถึงข้อมูลแบบพลวัต เพราะสามารถเข้าถึงข้อมูลได้ทั้งแบบสุ่มและแบบอนุบรรพ เช่น ถ้าข้อมูลที่ต้องการค้นหาเริ่มจากคีย์ 0010 ถึง 0100 และ 0200 ถึง 0500 และ 0300 ถึง 0400 ในรายงานเดียวกันเป็นต้น สำหรับการเข้าถึงข้อมูลแบบสุ่ม เหมาะกับข้อมูลที่มีลักษณะไม่แน่นอนอาจเรียงลำดับ หรือไม่เรียงลำดับคีย์ ที่ต้องการค้นหาอยู่กระจัดกระจายอยู่ทั่วไปภายในแฟ้มข้อมูลหลัก

ตัวอย่างเช่น การทำแฟ้มข้อมูลหลักพัสดุใช้งานให้เป็นปัจจุบัน เพื่อปรับปรุงชื่อพัสดุ และหน่วยนับให้ตรงกับแฟ้มข้อมูลหลักที่ 1 โดยนำทุกระเบียนในแฟ้มข้อมูลหลักพัสดุใช้งานซึ่งประกอบด้วยข้อมูล 50015 ซึ่งเรียงลำดับตามรหัสคลังและในแต่ละคลังเรียงลำดับตามรหัสพัสดุ ซึ่งรหัสพัสดุในแต่ละคลังย่อมจะมีโอกาสซ้ำกันบ้าง และรหัสพัสดุในแต่ละคลังอาจจะไม่ต่อเนื่องกันซึ่งอาจจะห่างกันเกินกว่าบีเอฟ ของแฟ้มข้อมูลนั้น ซึ่งดูเสมือนว่าคีย์ที่ต้องการค้นหาอยู่อย่างกระจัดกระจายภายในแฟ้มข้อมูลหลักที่ 1 ซึ่งประกอบด้วยข้อมูลถึง 249989 ระเบียนจึงสมควรจะเลือกการเข้าถึงข้อมูลแบบสุ่มจะเหมาะสมกว่าแบบอนุบรรพดังได้แสดงผลการทดลองเปรียบเทียบไว้ในบทที่ 5