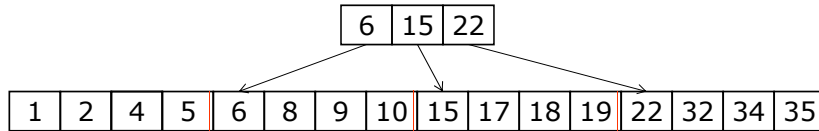


חיפוש רשומה

נתון קובץ ממוין בעל N רשומות ורוצים למצוא את הרשומה בעלת מפתח K

- פתרון נאיבי: סריקה סדרתית
 - ידרוש בממוצע קריאת $N/2$ רשומות, כלומר הבאת $N/2b$ בלוקים
 - חלוקת הקובץ לשני חלקים ושמירת הערכים המפרידים:
 - ידרוש בממוצע קריאת $N/4$ רשומות, כלומר $N/4b$ בלוקים
 - חלוקת הקובץ ל- m חלקים ושמירת הערכים המפרידים:
 - ידרוש בממוצע קריאת $N/2m$ רשומות, כלומר $N/2mb$ בלוקים

החלוקה הזו היא אינדקס לקובץ



אינדקסים

אינדקס

אינדקס הוא מבנה חיפוש אשר לכל ערך מפתח, מצביע לרשומה (רשומות) עם מפתח זה

- מאפשר גישה ישירה לרשומות בקובץ מובנה (יחס) גדול
- מקל על חישוב פעולות כמו בחירה, צירוף, חיתוך ועוד

למשל, חישוב $S \cap R$, אם יש אינדקס על R :
 לכל רשומה ב- S נבדוק באינדקס אם היא נמצאת ב- R
 – לפחות $|S|$ גישות לאינדקס של R ולרשומות ב- R (לאו דווקא בבלוקים)
 – אם S ממוין, אז ייתכן ששתי קריאות עוקבות תתייחסנה לאותו בלוק של R , ומספר הגישות יהיה קטן יותר

ישנם סוגים שונים של אינדקסים: עץ חיפוש (בינארי ו/או מאוזן), טבלת ערבול, ...

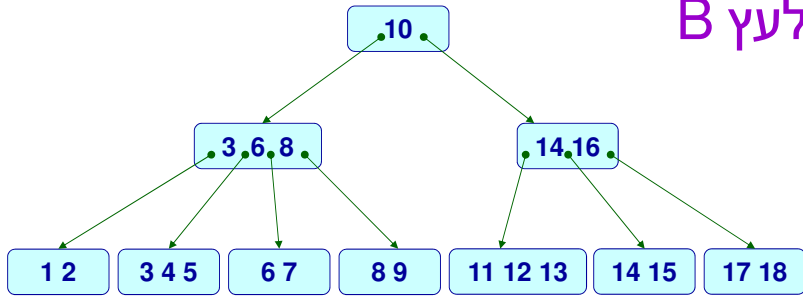
עבור קובץ גדול מאוד, גם האינדקס גדול מאוד וכולו או חלקו, נשמר בדיסק
 צריך לקחת בחשבון את עלות הקריאה מהדיסק (ולעבוד בבלוקים)

חיפוש רשומה

נתון קובץ ממוין בעל N רשומות ורוצים למצוא את הרשומה בעלת מפתח K

- חלוקת הקובץ ל- m חלקים ושמירת הערכים המפרידים:
 - ידרוש בממוצע קריאת $N/2m$ רשומות, כלומר הבאת $N/2mb$ בלוקים
- אבל אם m מאוד גדול אז הוא לא נכנס בזיכרון ויש למצוא את הרשומה בעלת מפתח K בקובץ ממוין בעל m רשומות
- חזרנו לאותה בעיה בקנה מידה יותר קטן:
 - נוכל להמשיך רקורסיבית עד ש- m יהיה קטן מספיק ← ייתן מבנה של עץ
 - נוכל לבחור את m בחכמה (כמה קטן?)
- מה נעשה עבור קובץ לא ממוין?

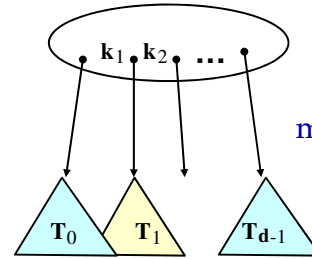
דוגמה לעץ B



עם $b=3, m=5$

האם זהו עץ B עבור $m = 3$? $m = 4$? $m = 7$?

עצי B (עם פרמטרים $m \geq 3, b$)



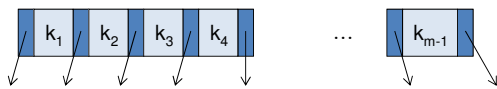
- ✓ לשורש דרגה בין 2 ו- m .
- ✓ לצומת פנימי (שאינו שורש) דרגה בין $\lceil m/2 \rceil$ ו- m
- ✓ כל העלים: באותו מרחק מהשורש
- ✓ הרשומות נמצאות רק בעלים,
- מספר הרשומות בעלה בין $\lceil b/2 \rceil$ ו- b
- ✓ מפתחות: בצומת פנימי עם d ילדים יש $d-1$ מפתחות,
- כך שהמפתח ה- i מקיים: $\max\{x \in T_{i-1}\} < k_i \leq \min\{x \in T_i\}$

גובה העץ: מספר הרמות לא כולל רמת העלים

גודל הצמתים הפנימיים והעלים

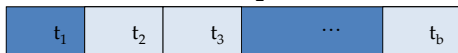
אם גודל מפתח הוא k וגודל מצביע הוא p , אז גודל צומת s_1 הוא

$$m p + (m-1) k = s_1$$



אם גודל רשומה הוא t , אז גודל עלה s_2 הוא

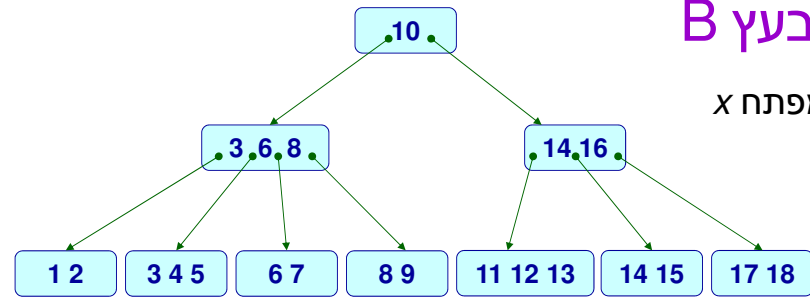
$$b t = s_2$$



הגדלת m, b מגדילה את הצמתים / עלים ומקטינה את גובה העץ (בדרך כלל, m, b הם בין 100 ל-200)

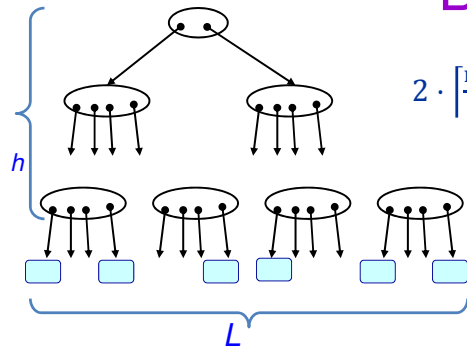
חיפוש בעץ B

מחפשים מפתח x



1. v מצביע לצומת, מאותחל לשורש
2. כל עוד v אינו מצביע לעלה,
- מצא את i שעבורו $v.k_i \leq x < v.k_{i+1}$
- (הילד ה- i של v) $v := \text{root of } T_i$
3. חפש את x בעלה v

גובה מקסימאלי של עץ B



$$2 \cdot \left\lceil \frac{m}{2} \right\rceil^{h_{\max}-1} \leq L$$

אם L ו- m גדולים

$$h_{\max} \leq 1 + \frac{\log_2 L - 1}{\log_2 m - 1} \approx 1 + \frac{\log_2 L}{\log_2 m}$$

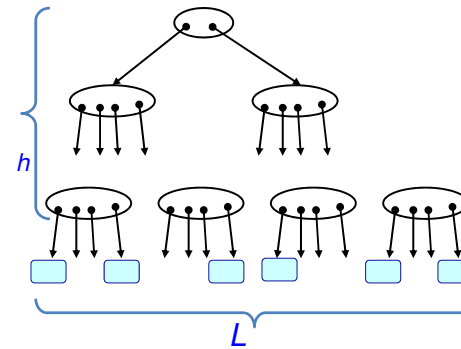
מאחר ו- h_{\max} מספר שלם:

$$h_{\max} \leq 1 + \left\lceil \frac{\log_2 L}{\log_2 m} \right\rceil = 1 + h_{\min}$$

$$h_{\max} - h_{\min} \leq 1 \Leftrightarrow$$

מקרא	
L	מספר עלים
L_{\min}	מספר עלים מינימלי
L_{\max}	מספר עלים מקסימלי
h	גובה
h_{\min}	גובה מינימלי
h_{\max}	גובה מקסימלי

גובה מינימאלי של עץ B



מספר העלים בעץ בגובה h

$$L_{\min} = 2 \lceil m/2 \rceil^{h-1} \quad \text{לפחות}$$

$$L_{\max} = m^h \quad \text{לכל היותר}$$

אם בעץ יש L עלים

$$m^{h_{\min}-1} < L \leq m^{h_{\min}}$$

$$\text{ולכן } h_{\min} = \lceil \log_m L \rceil = \left\lceil \frac{\log_2 L}{\log_2 m} \right\rceil$$

מקרא	
L	מספר עלים
L_{\min}	מספר עלים מינימלי
L_{\max}	מספר עלים מקסימלי
h	גובה
h_{\min}	גובה מינימלי
h_{\max}	גובה מקסימלי

דוגמה מספרית

קובץ בגודל 10 GB, עלים בגודל 75 KB (כל אחד)

$$m=200$$

$$L = 10 \text{ GB} / (75 \text{ KB} \times 0.69) = 201,650 \quad \text{מספר העלים הצפוי הוא}$$

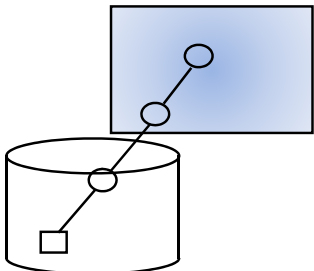
$$m \times 0.69 = 138 \quad \text{מספר הבנים הצפוי לכל צומת פנימי}$$

מספר הצמתים בכל רמה:

$$v_1 = L / (m \times 0.69) = 1462$$

$$v_2 = v_1 / (m \times 0.69) = 10$$

$$v_3 = 1 \quad \text{שורש העץ:}$$



זאת אומרת שגובה העץ הוא 3 וניתן לשמור את השורש ואת ילדיו בזיכרון הראשי (באופן קבוע)

גובה ממוצע

אם מתחילים בעץ ריק, ומבצעים רק הכנסות (בלי הוצאות)

וכל פרמוטציות הקלט שוות הסתברות

אפשר להוכיח שצפיפות המפתחות, בעלים ובצמתים הפנימיים:

$$E(\text{number of keys} / \text{maximal number of keys}) = \ln 2 \approx 0.69$$

$$\text{עבור } N \text{ מפתחות, מספר העלים הצפוי הוא } E(L) = \frac{N}{0.69b}$$

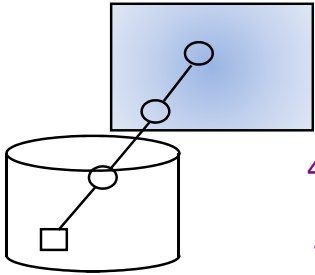
$$E(h) = \log_{0.69m} L = \log_{0.69m} \frac{N}{0.69b} = \frac{\log_2 N - \log_2 0.69b}{\log_2 0.69m}$$

$$= \frac{\log_2 N - \log_2 0.69 - \log_2 b}{\log_2 0.69 + \log_2 m} \approx \frac{0.54 + \log_2 N - \log_2 b}{-0.54 + \log_2 m}$$

מקרא	
L	מספר עלים
h	גובה

אם m גדול

המשך הדוגמה: זמן החיפוש בעץ-B



$$4 \text{ ms}/37 = 0.11 \text{ ms}$$

$$3.4 \text{ ms}$$

$$4 \text{ ms}/2 = 2 \text{ ms}$$

$$5.51 \text{ ms}$$

זמן קריאת צומת פנימי
 זמן העברת צומת פנימי
 זמן תזוזה (ממוצע)
 השהיית סיבוב (ממוצעת)
 סה"כ (בערך)

$$4 \text{ ms}/3 = 1.33 \text{ ms}$$

$$3.4 \text{ ms}$$

$$2 \text{ ms}$$

$$6.74 \text{ ms}$$

זמן קריאת עלה
 זמן העברת עלה
 זמן תזוזה (ממוצע)
 השהיית סיבוב (ממוצעת)
 סה"כ (בערך)

$$5.51 \text{ ms} + 6.74 \text{ ms} = 12.25 \text{ ms}$$

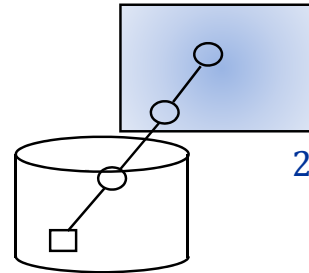
זמן קריאה כולל

המשך הדוגמה: זמן החיפוש בעץ-B

מעבר מהשורש אל העלה המתאים

עץ בגובה 3: השורש וילדיו בזיכרון הראשי, העלים והוריהם על הדיסק
 >דרגת העץ $m = 200$
 גודל מסילה: 225 KB

$$\text{גודל עלה: } 75 \text{ KB} \Leftarrow \text{במסילה יש } 3 = \frac{225 \text{ KB}}{75 \text{ KB}} \text{ עלים}$$



גודל מפתח 20B, גודל מצביע 10B

\Leftarrow גודל צומת פנימי

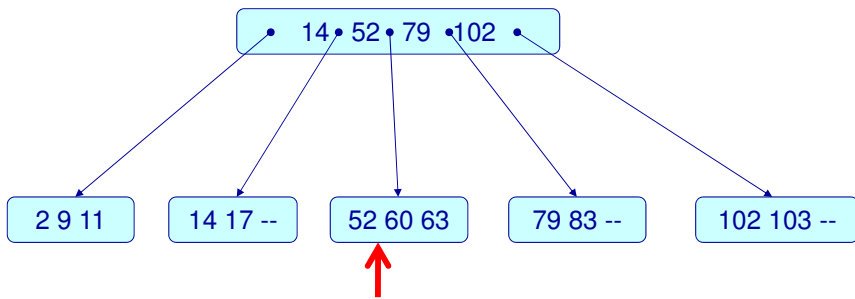
$$200 \times 10B + 199 \times 20B = 5980B \approx 6KB$$

$$\Leftarrow \text{במסילה יש } 37 = \left\lfloor \frac{225 \text{ KB}}{6 \text{ KB}} \right\rfloor \text{ צמתים פנימיים}$$

דוגמה להכנסה

$$m = 5, b = 3$$

הכנס רשומה עם מפתח 56



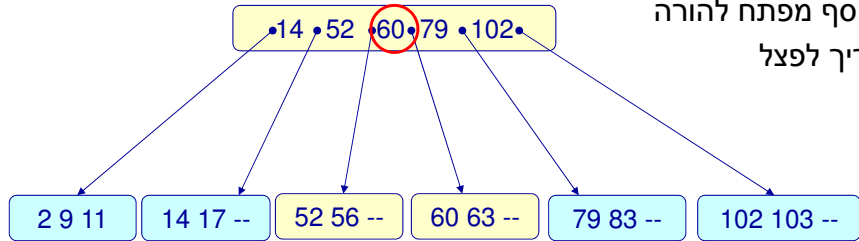
הכנסה לעץ B

1. מצא את העלה אליו צריך להוסיף
2. אם יש בו פחות מ b רשומות, הוסף את הרשומה
3. אחרת, יש לו $b+1$ רשומות, וצריך לפצל:
 - א. צור עלה חדש, והעבר אליו $\lfloor b/2 \rfloor + 1$ רשומות
 - ב. $\lfloor b/2 \rfloor$ רשומות נשארות בעלה הישן
 - ג. הוסף מפתח להורה
4. אם יש גלישה בצומת פנימי (יותר מ- m ילדים):
 פצל והוסף מפתח להורה
5. אם יש גלישה בשורש: פצל אותו לשנים וצור שורש חדש

דוגמה להכנסה

$m = 5, b = 3$

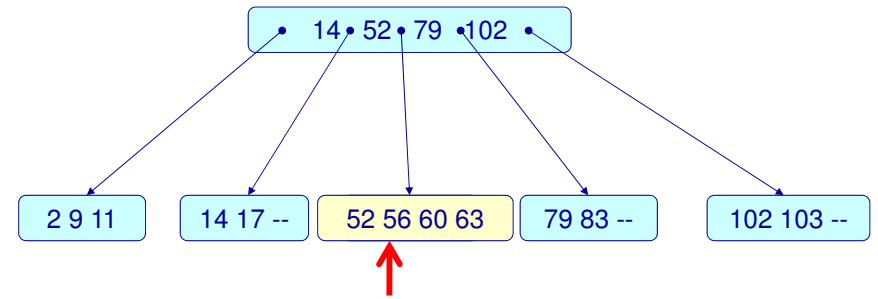
הכנס רשומה עם מפתח 56
צריך לפצל
הוסף מפתח להורה
צריך לפצל



דוגמה להכנסה

$m = 5, b = 3$

הכנס רשומה עם מפתח 56
צריך לפצל



זמן ההכנסה

אם אין פיצול, זמן ההכנסה הוא $T_{find+rotation} = 12.25ms + 4ms = 16.25ms$

מה ההסתברות לפיצול בסדרה של הכנסות? אם נוסף N רשומות לעץ ריק

במקרה הגרוע: מספר העלים הוא $N/(b/2) = 2N/b$

↔ מספר הפיצולים לכל היותר $2N/b$.

בממוצע: מספר העלים הצפוי הוא $L = N/(b \ln 2) = 1.44N/b$ וזה מספר הפיצולים הצפוי.

ההסתברות שבזמן הכנסה יקרה פיצול $L/N = 1.44/b$

לכן, עבור $b = 12$ ההסתברות פיצול = 12%

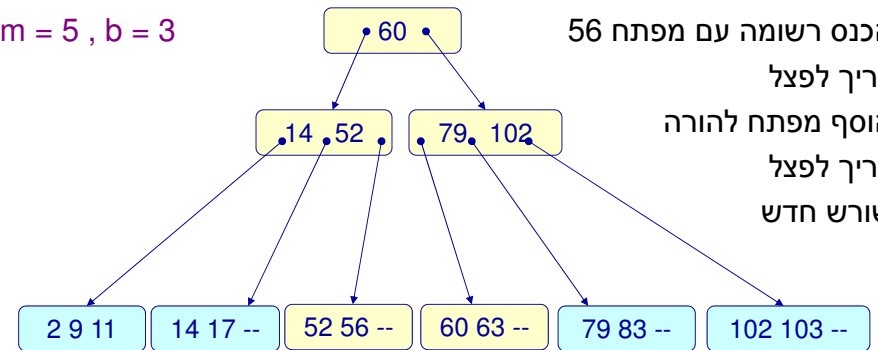
ועבור $b = 50$ ההסתברות פיצול = 3%

כיוון שזמן הפיצול דומה לזמן הכתיבה, ניתן להתעלם מזמן הפיצול (אם b גדול)

דוגמה להכנסה

$m = 5, b = 3$

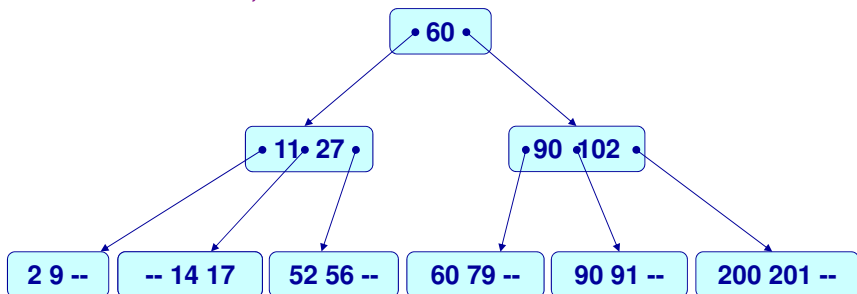
הכנס רשומה עם מפתח 56
צריך לפצל
הוסף מפתח להורה
צריך לפצל
שורש חדש



דוגמת הוצאה

$m = 5, b = 3$

הוצא את 11



הוצאת רשומה

1. חפש את הרשומה.
2. אם סילוקה לא יקטין את מספר הרשומות בעלה מתחת ל- $\lfloor b/2 \rfloor$ מחק את הרשומה וסיים.

אחרת:

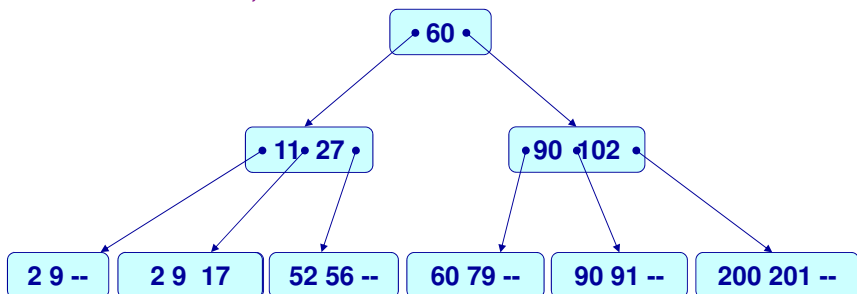
- אם לעלה של הרשומה יש אח שכן עם $\lfloor b/2 \rfloor + 1 \leq$ רשומות: העבר רשומות, עדכן את הצומת, את האח ואת המפתח המפריד אצל ההורה
 - אחרת: אָחַד את שני האחים, וסלק את המפתח המפריד מההורה, תוך שימוש באלגוריתם להוצאת רשומה מעלה.
3. אם לשורש נותר ילד יחיד: מחק את השורש, הילד הופך לשורש החדש.

כאשר מאחדים אחים: מספר הרשומות $\lfloor \frac{b}{2} \rfloor + \left(\lfloor \frac{b}{2} \rfloor - 1 \right) \leq \frac{b+1}{2} + \frac{b+1}{2} - 1 = b$ זמן הוצאה: דומה לזמן ההכנסה

דוגמת הוצאה

$m = 5, b = 3$

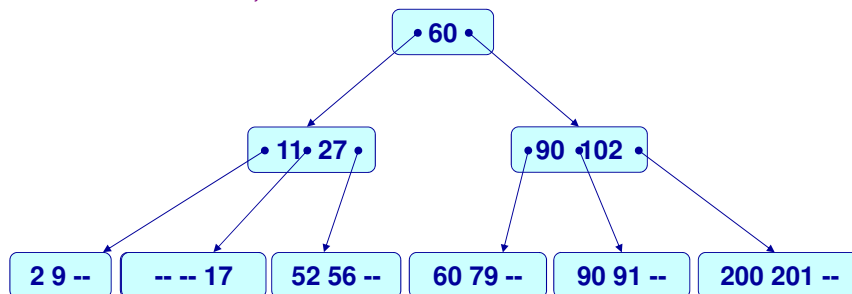
הוצא את 14: איחוד



דוגמת הוצאה

$m = 5, b = 3$

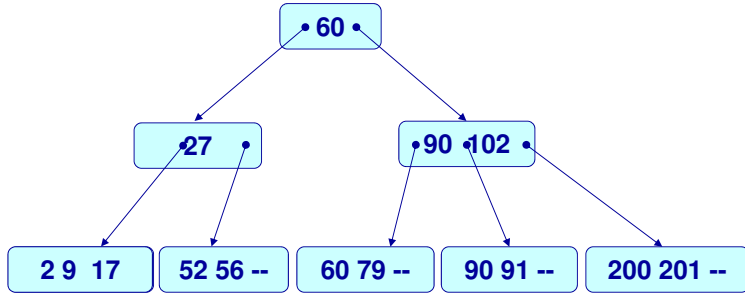
הוצא את 14



דוגמת הוצאה

$m = 5, b = 3$

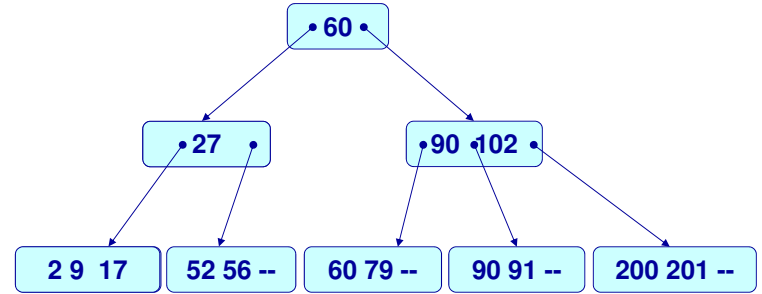
הוצא את 14: הסר את השורש



דוגמת הוצאה

$m = 5, b = 3$

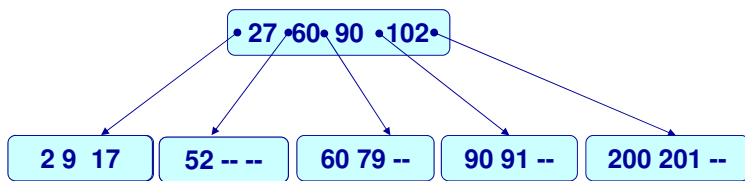
הוצא את 14: הוצא מפתח ברמה מעל



דוגמת הוצאה

$m = 5, b = 3$

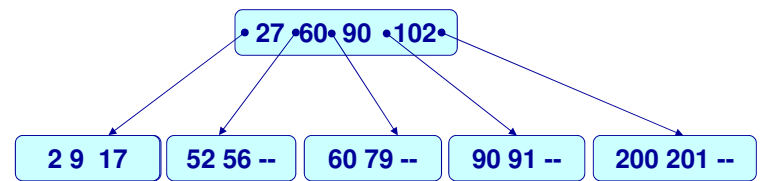
הוצא את 56



דוגמת הוצאה

$m = 5, b = 3$

הוצא את 14: הסר את השורש



מעבר סדרתי על עץ-B



נעריך את הזמן לקריאת כל העלים של קובץ בגודל 10 GB

$$75 \text{ KB} \times \ln 2 \cong 52 \text{ KB} \quad \text{בכל עלה}$$

$$10 \text{ GB} / 52 \text{ KB} = 201,650 \quad \text{מספר העלים}$$

$$T_{\text{leaf}} = 6.74 \text{ ms} \quad \text{זמן קריאת עלה}$$

$$6.74 \text{ ms} \times 201,650 = 1359.121 \text{ sec} = 22 \text{ min } 39 \text{ sec} \quad \text{סה"כ:}$$

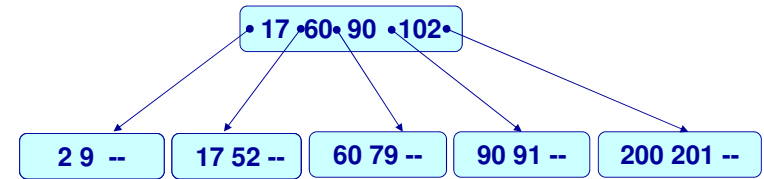
למה בחישוב זמן ההעברה של עלה הנחנו שהוא מלא, למרות שבממוצע הוא רק $\ln 2$ מלא?

מה היחס בין הזמן למעבר הסדרתי על עץ B לבין זמן העיבוד של קובץ סדרתי רציף?

דוגמת הוצאה

$$m = 5, b = 3$$

הוצא את 56 ואזן עלים



אם אין מקום לסבים בזיכרון ראשי? פתרון 2

נגדיל את דרגת הצמתים הפנימיים: נגדיל צמתים פנימיים ל-75 KB

אם גודל מפתח + מצביע הוא 30B

$$m = 75 \text{ KB} / 30 \text{ B} = 2560 \quad \text{דרגת העץ תהיה:}$$

$$(\ln 2)m \cong 0.69 \times 2560 = 1766 \quad \text{מספר ילדים ממוצע:}$$

אם בקובץ 201,650 עלים, מספר האבות $201,650 / 1766 = 114.2 \cong 115$

כל האבות הם בנים של השורש, ואותו אפשר להחזיק בזיכרון הראשי!

$$T_2 = 2 \times T_{\text{leaf}} = 2 \times 6.74 \text{ ms} = 13.48 \text{ ms} \quad \text{זמן החיפוש:}$$

אם אין מקום לסבים בזיכרון ראשי?

אפשר להחזיק 3 רמות בזיכרון המשני

זמן חיפוש:

$$T_1 = 2 \times T_{\text{internal}} + T_{\text{leaf}}$$

בדוגמה שלנו

$$T_1 = 2 \times 5.51 + 6.74 \text{ ms} = 17.76 \text{ ms}$$

רשומות עם כמה מפתחות

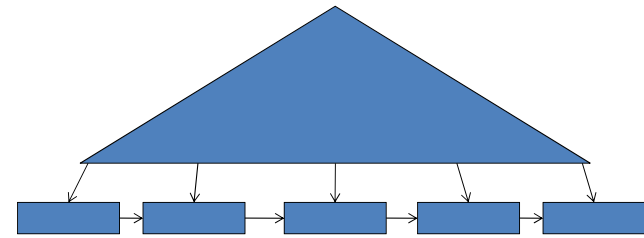
- נרצה לחפש רשומה גם לפי מס' ת"ז וגם לפי שם
- מפתח ראשי (מס' ת"ז)
- מפתח משני (שם)

אינדקס ראשי: הרשומות יסודרו לפי המפתח הראשי, למשל בעץ B
אינדקס משני: מבנה חיפוש נוסף לפי המפתח המשני

- לכל רשומה מקורית נבנה מיני-רשומה, המכילה את המפתח המשני, והפניה אל הרשומה המקורית במבנה הראשי
- את המיני-רשומות נארגן בקובץ במבנה כלשהו, למשל עץ B

שרשור של העלים

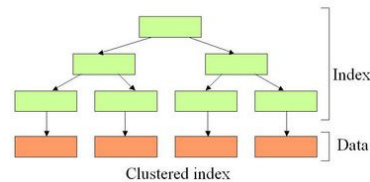
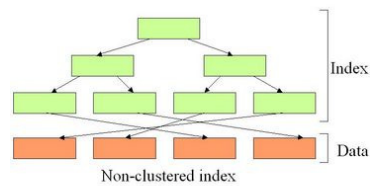
לעיתים, מוסיפים מצביעים: מכל עלה לעלה הבא אחריו מייעל קריאה של כל רשומות הקובץ באופן סדרתי על פי המפתח



מערכות קבצים

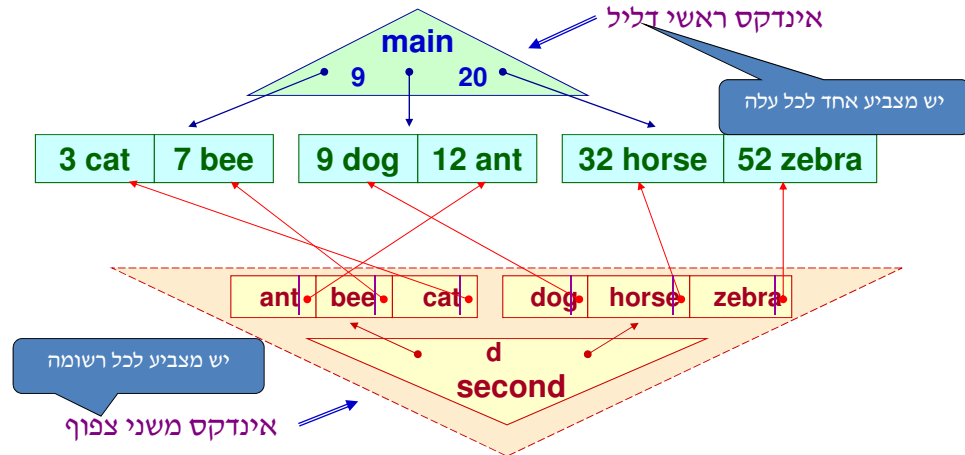
אינדקס מקובץ ואינדקס לא מקובץ

- **אינדקס מקובץ (clustered):** סדר הרשומות בקובץ תואם את סדר המפתחות באינדקס
 - זוג מפתחות סמוכים באינדקס צפויים להימצא באותו בלוק או בבלוקים סמוכים בדיסק
 - לעיתים קרובות אינדקס ראשי הוא אינדקס מקובץ
 - טוב לחיפוש טווח אבל מגדיל את העלות על עדכונים



- **אינדקס לא מקובץ (non-clustered):** סדר הרשומות לא תואם את סדר המפתחות
 - זוג מפתחות סמוכים באינדקס לא צפויים להימצא באותו בלוק או בבלוקים סמוכים בדיסק
 - אינדקס משני לרוב יהיה לא מקובץ

רשומות עם כמה מפתחות: דוגמה



אינדקס משני בלתי-נעוץ (unpinned)

במיני-רשומה (באינדקס המשני) יש את המפתח הראשי של הרשומה המקורית (המפתח הראשי צריך להיות ייחודי)

חיפוש (לפי מפתח משני):

- חפש באינדקס המשני, קבל מפתח ראשי k
- חפש את k באינדקס הראשי

הכנסה:

- הכנס את הרשומה החדשה בעזרת האינדקס הראשי
- הוסף לאינדקס המשני מיני-רשומה המכילה מפתח משני ומפתח ראשי

אינדקס משני נעוץ (pinned)

במיני-רשומה (באינדקס המשני) יש מצביע לעלה שבו הרשומה המקורית

חיפוש (לפי מפתח משני):

- חפש באינדקס המשני, קבל כתובת של עלה
- קרא את העלה וחפש את הרשומה בו

הכנסה:

- הכנס את הרשומה החדשה בעזרת האינדקס הראשי לעלה L
- הוסף לאינדקס המשני מיני-רשומה המכילה מפתח משני ומצביע ל-L
- אם העלה של האינדקס הראשי פוצל, עדכן את כל המפתחות באינדקס המשני שהצביעו לרשומות שנדדו לעלה אחר.
- האם לשמור מצביע לכתובת (עלה ומספר רשומה בעלה) של רשומה?

אינדקס משני משולב נעוץ / לא-נעוץ

במיני-רשומה שבאינדקס המשני שומרים את המפתח הראשי של הרשומה המקורית ומצביע למיקום (משוער) של הרשומה המקורית

חיפוש (לפי מפתח משני):

- חפש באינדקס המשני, קבל כתובת של עלה ומפתח ראשי k
- קרא את העלה וחפש את הרשומה בו
- אם לא נמצאה – חפש את k דרך האינדקס הראשי

עדכן את כתובת העלה באינדקס המשני

הכנסה:

- הכנס את הרשומה החדשה בעזרת האינדקס הראשי לעלה L
- הוסף לאינדקס המשני מיני-רשומה המכילה מפתח משני, מפתח ראשי מצביע ל-L

השוואה בין ארגון נעוץ ובלתי-נעוץ

חסרון של אינדקס נעוץ:

- כשמעבירים רשומות ממקום למקום בקובץ (למשל באיחוד או פיצול של עץ B) יש לעדכן את המיני-רשומות הרלוונטיות בכל האינדקסים המשניים
- יתכן שיש הרבה אינדקסים משניים (חלקם אפילו זמניים)

חסרון של אינדקס בלתי-נעוץ:

- זמן החיפוש כפול, כי מתוך האינדקס המשני מקבלים את המפתח הראשי ולא את מיקום הרשומה

דוגמה מספרית

נתון: קובץ עם 10^6 רשומות

כל רשומה 150 B

גודל הקובץ 150 MB

גודל עלה 75 KB

$75 \text{ KB} / 150 \text{ B} = 512$	מספר מקסימאלי של רשומות בעלה
$(\ln 2) 512 = 355$	מספר צפוי של רשומות בעלה
$10^6 / 355 = 2817$	מספר צפוי של עלים
6.74 ms	זמן קריאת עלה (חישבנו)

מעבר סדרתי לפי אינדקס ראשי: הגורם הדומיננטי הוא זמן קריאת העלים
 $2817 \times 6.74 = 18,987 \text{ ms} \approx 19 \text{ sec}$

יתרונות הפתרון המשולב

- לא מבזבזים זמן מיותר לעדכן את כל האינדקסים המשניים, כאשר מזיזים רשומה
- בחיפוש רשומה, מחיר כפול (או קצת יותר) רק בפעם הראשונה שניגשים אליה דרך האינדקס המשני לאחר העברתה
- אם יש כמה רשומות עם אותו מפתח משני ניתן לעתים לזהות את הרשומה המבוקשת ע"פ המפתח הראשי מבלי לבדוק את כל הרשומות האפשריות

זמן קריאת קובץ לפי אינדקס משני

☞ pinned (האינדקס המשני מכיל מצביעים לעלים):

מעבר על כל הקובץ רשומה אחר רשומה (סדרת הקריאות בדיסק היא אקראית): קריאת עלה (חישבנו) 6.74 msec

$$10^6 \times 6.74 \text{ ms} = 6,740 \text{ sec} \approx 1 \text{ hour } 52 \text{ min } 20 \text{ sec}$$

☞ unpinned (האינדקס המשני מכיל מפתחות של האינדקס הראשי):

יש צורך לבצע חיפוש לכל רשומה:
חיפוש רשומה (חישבנו) 12.25 ms

$$10^6 \times 12.25 \text{ ms} = 12,250 \text{ sec} \approx 3 \text{ hour } 24 \text{ min } 10 \text{ sec}$$

האם יש דרך יעילה יותר למעבר סדרתי לפי מפתח משני?

מעבר סדרתי לפי מפתח משני

ארגון לפי מצביעים לעלים:

$$30 \text{ B} = 10 \text{ B מצביע} + 20 \text{ B מפתח שבה}$$

• מספר מקסימאלי של מיני-רשומות בעלה: $75\text{KB}/30 \text{ B} = 2,560$

• מספר צפוי של מיני-רשומות בעלה: $(\ln 2) 2,560 = 1,775$

• מספר צפוי של עלי האינדקס המשני: $10^6/1775 = 564$

מספר זה זניח לעומת מספר הרשומות!

← הזמן הדומיננטי הוא זמן קריאת הרשומות עצמן!

צירוף $S(X, Y) \otimes R(Y, Z)$ באמצעות אינדקס

יש אינדקס צפוף של S עבור Y
לכל רשומה $r \in R$

חפש את $r.Y$ ב- S

עבור כל רשומה $s \in S$ שעבורה $s.Y = r.Y$
הוצא את $r \otimes s$ לפלט

– קריאת R : $B(R)$

– בממוצע לכל ערך של $r.Y$ יהיו $T(S)/V(S, Y)$ רשומות של S
(כאשר $V(S, Y)$ הוא מספר הערכים השונים של Y ב- S)

– סה"כ: $T(R) T(S)/V(S, Y)$
(כי הגישות ל S אינן לפי בלוקים)

שאלות טווח

בהינתן $x \leq y$, מצא את כל הרשומות בין x ל- y
דוגמה: עבור σ_C כאשר $C(x)$ הוא $a \leq x \leq b$
נקבל שאלת טווח.

שאלות טווח על אינדקס דליל יותר יעילות:

ניח k רשומות נמצאות בטווח, ובעלה יש (בממוצע) r רשומות

- אם השאלתה למפתח ראשי נעבור על k/r עלים
- אם השאלתה למפתח משני נעבור על k עלים