סיכום ונוסחאון במבני נתונים ומבוא לאלגוריתמים

קורס 20407 · עודכן ב-2026

(נכתב על ידי דניאל וונג – www.wongi.io)

כל ההגדרות, השגרות והמשפטים לפי ספר הקורס "מבוא לאלגוריתמים" (פרקים 1–14) ומדריך הלמידה, עם מספרי העמודים בספר. לכל נושא – תבנית פתרון כצעדים, דוגמה פתורה ושרטוט.

תוכן העניינים

נושא	עמוד	נושא	עמוד
ניתוח אסימפטוטי	1	מבני נתונים בסיסיים	6
נוסחאות נסיגה	2	טבלאות גיבוב	7
מיון וחיפוש	3	עצי חיפוש ועצים אדומים-שחורים	7
ערמות ותורי עדיפויות	5	הרחבת מבני נתונים	9

ביבליוגרפיה

קורמן, ת׳, לייזרסון, צ׳, ריבסט, ר׳ ושטיין, ק׳. *מבוא לאלגוריתמים*, פרקים 1–14 (מהדורה שנייה). רעננה: האוניברסיטה הפתוחה.

וינשטין, ג׳. *מבני נתונים ומבוא לאלגוריתמים – מדריך למידה* (20407). רעננה: האוניברסיטה הפתוחה.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. *Introduction to Algorithms* (2nd ed.). MIT Press.

1 ניתוח אסימפטוטי

1.1 ההגדרות

ניסוח המשפט	כתיב מתמטי
חסם עליון O – מ-n_0 והלאה, f לא עולה על g כפול קבוע. (עמ' 38–39)	$f(n) = O(g(n)) \iff \exists c > 0,\ n_0 :\ \forall n \geq n_0 :\ 0 \leq f(n) \leq c\,g(n)$
חסם תחתון Ω – מ-n_0 והלאה, f לפחות g כפול קבוע. (עמ' 39–40)	$f(n) = \Omega(g(n)) \iff \exists c > 0,\ n_0 :\ \forall n \geq n_0 :\ 0 \leq c\,g(n) \leq f(n)$
חסם הדוק Θ – גם O וגם Ω (משפט 3.1). (עמ' 37–40)	$c_1 g(n) \leq f(n) \leq c_2 g(n) \quad (n \geq n_0)$ $f = \Theta(g) \iff f = O(g)$ וגם $f = \Omega(g)$
o **ו-** ω **(לא הדוקים)** – לכל קבוע $c > 0$, ולא רק לאחד. נוח לבדוק בגבול. (עמ' 41–42)	$\lim \frac{f}{g} = 0 \Rightarrow f = o(g) \qquad \lim \frac{f}{g} = \infty \Rightarrow f = \omega(g)$ $0 < \lim \frac{f}{g} < \infty \Rightarrow f = \Theta(g)$
סדר הגדילה – מהאיטי למהיר. כל פולינום-לוג קטן מכל חזקה חיובית, וכל חזקה קטנה מכל מעריכית. (עמ' 43–47)	$1 \prec \lg^* n \prec \lg\lg n \prec \lg n \prec \lg^k n \prec n^\varepsilon \prec n \prec n\lg n \prec n^2 \prec 2^n \prec n! \prec n^n$
כללי לוגריתמים וסכומים שימושיים. (עמ' 44–45, נספח א)	$a^{\log_b n} = n^{\log_b a} \qquad \log_b n = \frac{\lg n}{\lg b} \qquad \lg(n!) = \Theta(n\lg n)$ $\sum_{i=0}^{k-1} 2^i = 2^k - 1 \qquad \sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ $\sum_{i=1}^{n} \frac{1}{i} = \Theta(\lg n)$

1.2 הוכחה או הפרכה לפי ההגדרה

אלגוריתם להוכחה או להפרכה לפי ההגדרה (עמ' 38–39):

1. כתוב את מה שצריך להוכיח **לפי ההגדרה**: אילו קבועים צריך למצוא (c_1, c_2, n_0) ואיזה אי-שוויון צריך להתקיים.
2. **חלק** ב-n בחזקה הגבוהה, כדי לקבל אי-שוויון בקבועים בלבד.
3. **בחר קבועים מפורשים**: c_2 מהצד העליון, ואחר כך n_0 ו-c_1 מהצד התחתון.
4. אם יש **נתונים** (למשל $f = O(g)$) – כתוב כל אחד עם קבועים **משלו** (c_1, n_1) ו-(c_2, n_2), וקח $n_0 = \max(n_1, n_2)$.
5. **להפרכה:** הנח שקיימים קבועים, והגע לסתירה – אי-שוויון שנכשל לכל n גדול מספיק.

לדוגמה נתון (הספר, עמ' 38–39): להוכיח $\frac{1}{2}n^2 - 3n = \Theta(n^2)$, ולהפריך $6n^3 = \Theta(n^2)$.

נפעיל את האלגוריתם כדלקמן:

$$c_1 n^2 \leq \tfrac{1}{2}n^2 - 3n \leq c_2 n^2 \quad \xrightarrow{:n^2} \quad c_1 \leq \tfrac{1}{2} - \tfrac{3}{n} \leq c_2$$

$$n_0 = 7 \Rightarrow \tfrac{1}{2} - \tfrac{3}{7} = \tfrac{1}{14} = c_1\ ;\ (n \geq 1 \text{ לכל}) \ c_2 = \tfrac{1}{2}$$

הפרכה: $6n^3 \leq c_2 n^2 \iff n \leq c_2/6$ – נכשל לכל $n > c_2/6$

נקבל אפוא את המבוקש:

$$6n^3 \neq \Theta(n^2)\ ;\ c_1 = \tfrac{1}{14},\ c_2 = \tfrac{1}{2},\ n_0 = 7 \text{ עם } \tfrac{1}{2}n^2 - 3n = \Theta(n^2)$$

1.3 נכונות אלגוריתם – שמורת לולאה

ניסוח המשפט	כתיב מתמטי
פסאודו-קוד בספר – $\leftarrow$ השמה; $=$ גוש; $\triangleright$ הערה; $A[i..j]$ תת-מערך; שדות $f[x]$; פרמטרים מועברים לפי ערך; ההשוואות *and*, *or* מקוצרות. (עמ' 16–18)	$key[x] \qquad length[A] \qquad i \leftarrow j \leftarrow e$

אלגוריתם להוכחת נכונות בעזרת שמורת לולאה (עמ' 15–16):

1. **נסח שמורה** – טענה על המשתנים שנכונה **בתחילת כל איטרציה**.
2. **אתחול:** השמורה נכונה לפני האיטרציה הראשונה.
3. **תחזוקה:** אם היא נכונה לפני איטרציה – היא נכונה גם לפני הבאה.
4. **סיום:** כשהלולאה נגמרת, השמורה (עם תנאי היציאה) נותנת את הנכונות.

לדוגמה נתון (הספר, עמ' 15–16): INSERTION-SORT, הלולאה החיצונית על $j = 2..n$.

נפעיל את האלגוריתם כדלקמן:

שמורה: בתחילת כל איטרציה, $A[1..j-1]$ מכיל את האיברים המקוריים של $A[1..j-1]$, **ממוינים**

אתחול – $j = 2$: $A[1]$ לבד ממוין; תחזוקה – $A[j]$ מוכנס למקומו; סיום – $j = n+1$

נקבל אפוא את המבוקש:

$A[1..n]$ ממוין – האלגוריתם נכון

2 נוסחאות נסיגה

2.1 שלוש השיטות

ניסוח המשפט	כתיב מתמטי
שיטת ההצבה – מנחשים את התשובה ומוכיחים באינדוקציה (עם אותו קבוע c בהנחה ובמסקנה!). אם הצעד "כמעט" יוצא – מחסירים איבר מסדר נמוך מהניחוש. (עמ' 53–57)	ניחוש: $T(n) \le cn\lg n$ $T(n) \le 2c\frac{n}{2}\lg\frac{n}{2} + n = cn\lg n - cn + n \le cn\lg n \ (c \ge 1)$
עץ רקורסיה / איטרציה – פותחים את הנוסחה, סוכמים את העלות בכל רמה, וסוכמים את הרמות. טוב לניחוש, ואז מוכיחים בהצבה. (עמ' 58–62)	$T(n) = 3T(n/4) + cn^2 \Rightarrow \sum_i \left(\frac{3}{16}\right)^i cn^2 = \Theta(n^2)$
החלפת משתנים – נוסחה עם $\sqrt{n}$ (או $n^{1/b}$): מסמנים $m = \lg n$, ו-$S(m) = T(2^m)$; פותרים את S וחוזרים ל-T. (עמ' 57; מדריך, עמ' 40)	$T(n) = 2T(\sqrt{n}) + \lg n \xrightarrow{m=\lg n} S(m) = 2S(m/2) + m$ $S(m) = O(m\lg m) \Rightarrow T(n) = O(\lg n \lg\lg n)$
שיטת האב (משפט 4.1) – רק לצורה $aT(n/b) + f(n)$, $a \ge 1,\ b > 1$. משווים את $f(n)$ ל-$n^{\log_b a}$. (עמ' 63–65)	1. $f = O(n^{\log_b a - \varepsilon}) \Rightarrow T = \Theta(n^{\log_b a})$ 2. $f = \Theta(n^{\log_b a}) \Rightarrow T = \Theta(n^{\log_b a}\lg n)$ 3. $f = \Omega(n^{\log_b a + \varepsilon})$ וגם $af(n/b) \le cf(n),\ c<1 \Rightarrow T = \Theta(f(n))$
שיטת האב המורחבת (תרגיל 4.4-2) – כש-$f(n)$ גדולה מ-$n^{\log_b a}$ רק בפקטור לוגריתמי, שיטת האב הרגילה לא חלה (עמ' 64); מקרה 2 המורחב פותר זאת. (מדריך, עמ' 44)	$f(n) = \Theta(n^{\log_b a}\lg^k n),\ k \ge 0 \Rightarrow T(n) = \Theta(n^{\log_b a}\lg^{k+1} n)$ $2T(n/2) + n\lg n = \Theta(n\lg^2 n)$
נוסחאות מהספר: חיפוש בינרי, מיון מיזוג, המקרה הגרוע של מיון מהיר, ושלוש הדוגמאות לשיטת האב. (עמ' 31, 28, 125, 64)	$2T(n/2) + \Theta(n) = \Theta(n\lg n)$ $T(n/2) + \Theta(1) = \Theta(\lg n)$ $9T(n/3) + n = \Theta(n^2)$ $T(n-1) + \Theta(n) = \Theta(n^2)$ $3T(n/4) + n\lg n = \Theta(n\lg n)$ $T(2n/3) + 1 = \Theta(\lg n)$

2.2 נוסחה עם $n-1$ (שיטת האב לא חלה)

אלגוריתם לפתרון $T(n)=aT(n-1)+f(n)$ (פתיחה ואינדוקציה):

1. **פתח** את הנוסחה 2–3 פעמים (עץ רקורסיה עם ענף אחד בכל רמה), ורשום את הצורה אחרי k הצבות: $a^kT(n-k)+\sum_{i=0}^{k-1}a^if(n-i)$.
2. **בחר** k שמביא לתנאי ההתחלה ($k=n-1$ עבור $T(1)$).
3. **חשב** את הסכום (טור חשבוני / הנדסי – נספח א).
4. **הוכח** את הנוסחה הסגורה באינדוקציה (שיטת ההצבה, עמ' 53), ורק אז כתוב Θ.

לדוגמה נתון (הספר, עמ' 125 – המקרה הגרוע של מיון מהיר): $T(n)=T(n-1)+n$, $T(1)=1$.

נפעיל את האלגוריתם כדלקמן:

$$T(n)=T(n-k)+\big(n+(n-1)+\cdots+(n-k+1)\big) \xrightarrow{k=n-1} 1+2+\cdots+n$$

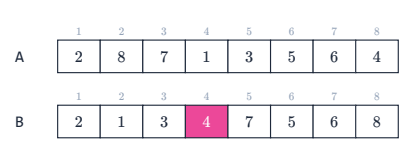
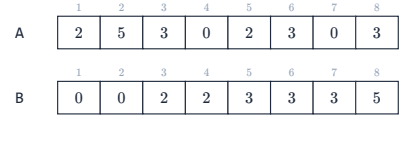
אינדוקציה: $\frac{(n-1)n}{2}+n=\frac{n(n+1)}{2}$ ✓

נקבל אפוא את המבוקש:

$$T(n)=\frac{n(n+1)}{2}=\Theta(n^2)$$

3 מיון וחיפוש

3.1 אלגוריתמי המיון

ניסוח המשפט	שרטוט	כתיב מתמטי
מיון הכנסה – יציב, במקום. $\Theta(n)$ על מערך ממוין. (עמ' 14–20)		$\Theta(n^2)$ גרוע וממוצע
מיון בחירה (תרגיל 2.2-2) – בכל סבב מוצאים את הקטן ביותר ב-$A[j..n]$ ומחליפים אותו עם $A[j]$; $n-1$ סבבים. (עמ' 23)		$\Theta(n^2)$ בכל מקרה
מיון מיזוג – יציב, **לא** במקום ($\Theta(n)$ זיכרון). MERGE של שני מערכים ממוינים – $\Theta(n)$. (עמ' 24–28)		$T(n)=2T(n/2)+\Theta(n)=\Theta(n\lg n)$
מיון ערמה – במקום, **לא** יציב. BUILD-MAX-HEAP ואז $n-1$ פעמים החלפה עם השורש ו-MAX-HEAPIFY. (עמ' 113)		$O(n)+(n-1)\cdot O(\lg n)=\Theta(n\lg n)$
מיון מהיר – במקום, לא יציב. PARTITION סביב הציר $x=A[r]$: משמאל ל-q – קטנים או שווים, מימין – גדולים; ואז רקורסיה על שני הצדדים. גרוע – חלוקה $n-1:0$ (למשל מערך ממוין): $\Theta(n^2)$; חלוקה מאוזנת (גם $1:9$) – $\Theta(n\lg n)$; RANDOMIZED-QUICKSORT בוחר ציר אקראי. (עמ' 122–130)	 PARTITION (איור 7.1): $x=4\Rightarrow q=4$	
חסם תחתון למיון השוואות (משפט 8.1) – עץ ההחלטה הוא בעל $n!$ עלים לפחות, ולכן גובהו $\Omega(n\lg n)$. **כל** מיון השוואות – ערמה, מיזוג, מהיר – לא יכול לרדת מזה. (עמ' 137–139)		$h\geq\lg(n!)=\Omega(n\lg n)$
מיון מנייה – מספרים שלמים בתחום $0..k$: $C[i]$ = כמה פעמים i; מצטברים ($C[i]$ = כמה $\leq i$); מציבים מהסוף $B[C[A[j]]]\leftarrow A[j]$ – **יציב**. $\Theta(n+k)$, ואם $k=O(n)$: $\Theta(n)$. (עמ' 140–141)	 $C=\langle2,0,2,3,0,1\rangle\rightarrow\langle2,2,4,7,7,8\rangle$ (איור 8.2)	
מיון בסיס (למה 8.3) – d ספרות, כל ספרה מיון יציב (מנייה) מהספרה **הפחות** משמעותית. מספרים עד n^c בבסיס n: c ספרות. (עמ' 142–144)		$\Theta(d(n+k))$ בבסיס $[0,n^c)$ n: $\Theta(c(n+n))=\Theta(n)$

ניסוח המשפט	שרטוט	כתיב מתמטי
מיון דליים – קלט מפוזר אחיד ב-$[0,1)$: n דליים ($A[i]$ לדלי $\lfloor nA[i] \rfloor$), כל דלי במיון הכנסה, ושרשור. (עמ' 145–146)		ממוצע $\Theta(n)$
סטטיסטי סדר – המינימום ב-$n-1$ השוואות; RANDOMIZED-SELECT – $\Theta(n)$ בממוצע; SELECT (חציון של חציונים בחמישיות) – $\Theta(n)$ במקרה הגרוע. (עמ' 153–159)		$T(n) \le T(\lceil n/5 \rceil) + T(7n/10+6) + O(n) = O(n)$
חיפוש בינרי – במערך ממוין, חוצים כל פעם (תרגיל 2.3-5). (עמ' 31)		$T(n) = T(n/2) + \Theta(1) = \Theta(\lg n)$

השוואת המיונים (פרקים 2, 6–8; יציבות – תרגיל 8.3-2, עמ' 144)

מיון	מקרה גרוע	מקרה ממוצע	מקרה טוב	במקום	יציב	עמ'
הכנסה	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n)$	כן	כן	20–14
בחירה	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	כן	לא	23
מיזוג	$\Theta(n \lg n)$	$\Theta(n \lg n)$	$\Theta(n \lg n)$	לא	כן	28–24
ערמה	$O(n \lg n)$	$O(n \lg n)$	$O(n \lg n)$	כן	לא	113
מהיר	$\Theta(n^2)$	$\Theta(n \lg n)$	$\Theta(n \lg n)$	כן	לא	130–122
מנייה	$\Theta(n+k)$	$\Theta(n+k)$	$\Theta(n+k)$	לא	כן	141–140
בסיס	$\Theta(d(n+k))$	$\Theta(d(n+k))$	$\Theta(d(n+k))$	לא	כן	144–142
דליים	$\Theta(n^2)$	$\Theta(n)$	$\Theta(n)$	לא	כן	146–145

3.2 מיון בזמן לינארי – מתי ואיך

אלגוריתם לבחירת שיטת מיון לפי הזמן המבוקש (פרק 8):

1. $\Theta(n \lg n)$? מיון השוואות – ערמה או מיזוג (מהיר – רק בממוצע).
2. $\Theta(n)$? אסור מיון השוואות (משפט 8.1, עמ' 137) – בדוק אם המספרים **שלמים בתחום חסום**:
   - תחום k עם $k = O(n)$ – **מיון מנייה** (עמ' 140).
   - תחום עד $n^c - 1$ – **מיון בסיס** בבסיס n: כל מספר הוא c ספרות בתחום $0..n-1$, וכל ספרה – מיון מנייה (למה 8.3, עמ' 143).
   - מספרים ממשיים בפיזור אחיד ב-$[0,1)$ – **מיון דליים**, $\Theta(n)$ בממוצע (עמ' 145).
3. **הנתונים כבר בעץ חיפוש?** סריקה תוכית מחזירה אותם ממוינים ב-$\Theta(n)$ (משפט 12.1, עמ' 214–215).
4. **שתי סדרות ממוינות?** MERGE – מעבר אחד במקביל, $\Theta(n)$ (עמ' 24).

לדוגמה נתון (תרגיל 8.3-4, עמ' 144): למיין n שלמים בתחום $0..n^2-1$ בזמן $O(n)$. כאן $n = 8$:

	1	2	3	4	5	6	7	8
A	45	7	63	22	9	30	51	0

נפעיל את האלגוריתם כדלקמן:

$\Longrightarrow n^2 - 1 = 63$ בבסיס 8 כל מספר הוא **2 ספרות**: $45 = (5,5)_8,\ 7 = (0,7)_8,\ 63 = (7,7)_8, \ldots$

מיון מנייה לפי הספרה **הנמוכה** (יציב): $0,\ 9,\ 51,\ 45,\ 22,\ 30,\ 7,\ 63$

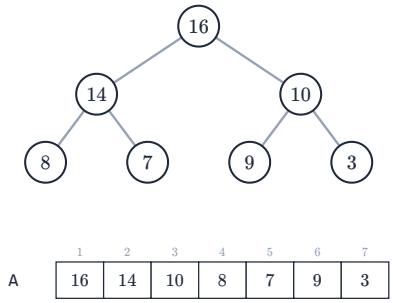
מיון מנייה לפי הספרה **הגבוהה**: $0,\ 7,\ 9,\ 22,\ 30,\ 45,\ 51,\ 63$

נקבל אפוא את המבוקש:

$$\Theta(d(n+k)) = \Theta(2(n+n)) = \Theta(n)$$

4 ערמות ותורי עדיפויות

4.1 ערמה בינרית

ניסוח המשפט	שרטוט	כתיב מתמטי
ערמה במערך – עץ כמעט שלם; הילדים וההורה לפי האינדקס: $\text{PARENT}(i) = \lfloor i/2 \rfloor$, $\text{LEFT}(i) = 2i$, $\text{RIGHT}(i) = 2i + 1$. **ערמת מקסימום:** כל הורה $\geq$ ילדיו, ולכן המקסימום בשורש. (עמ' 106–107)		
גובה ועלים – גובה n איברים הוא $\lfloor \lg n \rfloor$. העלים הם $A[\lfloor n/2 \rfloor + 1..n]$. (עמ' 107–108, תרגיל 6.1-7)		$h = \lfloor \lg n \rfloor \quad \leq \lceil n/2^{h+1} \rceil$ צמתים בגובה h
MAX-HEAPIFY – מוריד את $A[i]$ למטה, החלפה עם הילד **הגדול**. (עמ' 108–109)		$O(h) = O(\lg n)$
BUILD-MAX-HEAP – MAX-HEAPIFY מ-$\lfloor n/2 \rfloor$ עד 1. לינארי, כי רוב הצמתים נמוכים. (עמ' 110–112)		$\sum_h \frac{n}{2^{h+1}} O(h) = O(n)$
תור עדיפויות – HEAP-MAXIMUM $\Theta(1)$; HEAP-EXTRACT-MAX (האחרון לשורש ו-MAX-HEAPIFY); HEAP-INCREASE-KEY (מעלים למעלה עם ההורה); MAX-HEAP-INSERT (מוסיפים $-\infty$ בסוף ומגדילים); HEAP-DELETE (תרגיל 6.5-7) – האחרון במקום $A[i]$, ואז מתקנים למעלה או למטה. ערמת מינימום – הכול הפוך. (עמ' 116–118)		$O(\lg n)$ לכל פעולה (חוץ מ-MAXIMUM)

4.2 טבלת יאנג (בעיה 6-3)

ניסוח המשפט	שרטוט	כתיב מתמטי
טבלת יאנג $n \times m$ – כל שורה ממוינת משמאל לימין, כל עמודה מלמעלה למטה; משבצת ריקה – ∞. המינימום ב-$Y[1,1]$; אם $Y[1,1] = \infty$ הטבלה ריקה, ואם $Y[m,n] < \infty$ היא מלאה (סעיפים א–ב). בשרטוט – הטבלה של סעיף א. (עמ' 119–120)	$\begin{array}{\|c\|c\|c\|c\|}\hline 2 & 3 & 14 & 16 \\ \hline 4 & 8 & \infty & \infty \\ \hline 5 & 12 & \infty & \infty \\ \hline 9 & \infty & \infty & \infty \\ \hline \end{array}$	

אלגוריתם לשלוש הפעולות על טבלת יאנג, כל אחת $O(m+n)$ (בעיה 3-6, סעיפים ג, ד, ו):

1. **חיפוש K (סעיף ו):** התחל ב**פינה הימנית-העליונה** $Y[1,n]$ – הגדולה בשורה שלה והקטנה בעמודה שלה.
   - $Y[i,j]=K$ – מצאת.
   - $Y[i,j]>K$ – כל העמודה מכאן ולמטה גדולה מ-K: זוז **שמאלה**.
   - $Y[i,j]<K$ – כל השורה מכאן ושמאלה קטנה מ-K: רד **למטה**.
   - יצאת מהטבלה – K לא קיים. כל צעד מוחק שורה או עמודה: לכל היותר $m+n-1$ בדיקות.
2. **הכנסה (סעיף ד):** שים את K ב-$Y[m,n]$ (שהוא ∞ בטבלה לא מלאה), **והחלף עם הגדול** מבין השכן שמעל והשכן משמאל, כל עוד הוא גדול מ-K.
3. **EXTRACT-MIN (סעיף ג):** החזר את $Y[1,1]$, שים שם ∞, **והחלף עם הקטן** מבין השכן שמימין והשכן שמתחת, כל עוד הוא קטן ממנו – כמו MAX-HEAPIFY (עמ' 108). נוסחת הנסיגה: $T(p)=T(p-1)+O(1)=O(p)$, כאשר $p=m+n$.
4. **למה זה נכון:** בכל החלפה האיבר שזז נכנס למקום שבו הוא מתאים לשכניו החדשים, והבעיה היחידה שנשארת היא סביב K (או סביב ה-∞) – וזה מה שהלולאה בודקת שוב. כל החלפה מקטינה (או מגדילה) i או j: לכל היותר $m+n-2$ החלפות.

לדוגמה נתון (בעיה 3-6, סעיף א): הטבלה של $\{9,16,3,2,4,8,5,14,12\}$. נחפש את 12, נכניס 7 ונבצע EXTRACT-MIN.

נפעיל את האלגוריתם כדלקמן:

חיפוש 12: 14, 16 שמאלה, 8, 3 למטה – נמצא ב-$(3,2)$

2	3 ←	14 ←	16
4	8 ↓	∞	∞
5	12	∞	∞
9	∞	∞	∞

הכנסת 7: מ-$(4,4)$ למעלה ושמאלה, עד $(1,3)$

2	3	7	16
4	8	14 ←	∞
5	12	∞	∞
9	∞	∞	∞

EXTRACT-MIN: מחזיר 2; ה-∞ יורד ל-$(3,2)$

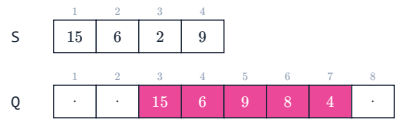
3 →	8 ↓	14	16
4	12	∞	∞
5	∞	∞	∞
9	∞	∞	∞

נקבל אפוא את המבוקש:

12 נמצא אחרי 5 בדיקות; 7 עולה ב-4 החלפות; 3 – EXTRACT-MIN החלפות. בכל אחת $\leq m+n-1=7$

5 מבני נתונים בסיסיים

5.1 מחסנית, תור ורשימה מקושרת

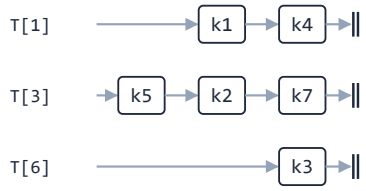
ניסוח המשפט	שרטוט	כתיב מתמטי
מחסנית (LIFO) – PUSH, POP ב-$O(1)$; $top[S]$; POP על מחסנית ריקה – חמיקה (underflow). **תור** (FIFO) – ENQUEUE, DEQUEUE ב-$O(1)$, מערך מעגלי עם *head* (הראשון) ו-*tail* (המקום הפנוי הבא). (עמ' 167–170)	 $top[S]=4$ $head[Q]=3$, $tail[Q]=8$ (כמו באיורים 10.1, 10.2)	
רשימה מקושרת – LIST-SEARCH $\Theta(n)$; LIST-INSERT (בראש הרשימה) $O(1)$; LIST-DELETE $O(1)$ בדו-כיוונית כשנתון מצביע לאיבר. (עמ' 171–174)		
זקיף $nil[L]$ – צומת דמה בין הראש לזנב ברשימה מעגלית; חוסך את בדיקות המקרים בקצוות. (עמ' 173–174)		$next[nil[L]]=head$, $prev[nil[L]]=tail$

5.2 מימוש מצביעים ועצים בעלי שורש

ניסוח המשפט	כתיב מתמטי
מימוש מצביעים במערכים – שלושה מערכים *next*, *key*, *prev* (או מערך יחיד עם היסטים); המצביע הוא אינדקס. **רשימת פנויים** – מחסנית של המקומות הפנויים: ALLOCATE-OBJECT ו-FREE-OBJECT ב-$O(1)$. (עמ' 175–179)	$O(1)$ להקצאה ולשחרור

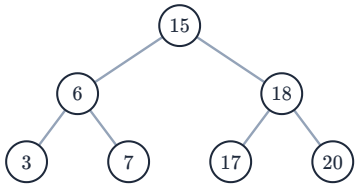
ניסוח המשפט	כתיב מתמטי
עצים בעלי שורש – בינרי: p, $left$, $right$. מספר בנים לא חסום: **בן שמאלי, אח ימני** – כל צומת מצביע לבן השמאלי ביותר שלו ולאח שמימינו; $O(n)$ זיכרון לכל עץ. (עמ' 179–181)	$left\text{-}child[x]$, $right\text{-}sibling[x]$, $p[x]$

6 טבלאות גיבוב

ניסוח המשפט	שרטוט	כתיב מתמטי
מיעון ישיר – מערך בגודל התחום U; $T[k]$. רק כשהתחום קטן. (עמ' 187–188)		$O(1)$ לכל פעולה, $\Theta(\lvert U\rvert)$ זיכרון
שרשור – כל משבצת רשימה מקושרת. **מקדם העומס** $\alpha = n/m$. בהנחת פיזור אחיד פשוט, חיפוש (מוצלח או לא) – $\Theta(1+\alpha)$ בממוצע (משפטים 11.1, 11.2). (עמ' 189–192)	 $\Theta(1+\alpha)$ בממוצע; $\alpha = \frac{n}{m}$ גרוע: $\Theta(n)$	
שיטת החילוק – m ראשוני רחוק מחזקה של 2. **שיטת הכפל** – $0 < A < 1$, למשל $A \approx (\sqrt{5}-1)/2$; m לא משנה. (עמ' 194–195)		$h(k) = k \bmod m$ $h(k) = \lfloor m\,(kA \bmod 1)\rfloor$
מיעון פתוח – הכול בטבלה ($\alpha < 1$); סדרת בדיקות $h(k,0), h(k,1), \ldots$. **ליניארי** (הצטברות ראשונית), **ריבועי**, **גיבוב כפול** (הכי טוב). מחיקה – מסמנים DELETED. (עמ' 200–203)		$h(k,i) = (h'(k)+i) \bmod m$ $h(k,i) = (h'(k)+c_1 i + c_2 i^2) \bmod m$ $h(k,i) = (h_1(k) + i\,h_2(k)) \bmod m$
ניתוח מיעון פתוח (גיבוב אחיד) – חיפוש לא מוצלח / הכנסה (משפט 11.6); חיפוש מוצלח (משפט 11.8). (עמ' 203–205)		$\leq \frac{1}{1-\alpha}$ $\leq \frac{1}{\alpha}\ln\frac{1}{1-\alpha}$

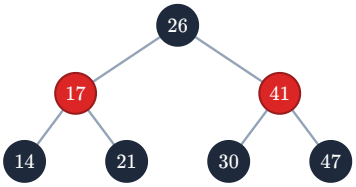
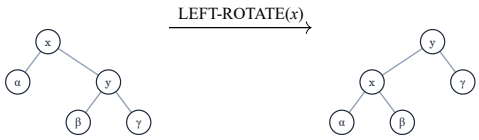
7 עצי חיפוש ועצים אדומים-שחורים

7.1 עץ חיפוש בינרי

ניסוח המשפט	שרטוט	כתיב מתמטי
תכונת עץ החיפוש – בתת-העץ השמאלי $\leq key[x] \leq$ בתת-העץ הימני. **סריקה תוכית** (משפט 12.1) מדפיסה ממוין ב-$\Theta(n)$. (עמ' 213–215)	 תוכית: $3, 6, 7, 15, 17, 18, 20$	
שלוש הסריקות – **תחילית:** השורש, ואז שמאל וימין; **תוכית:** שמאל, השורש, ימין; **סופית:** שמאל, ימין, ואז השורש. כל אחת $\Theta(n)$. בעץ שבשרטוט למעלה: (עמ' 214–215)		תחילית: $15, 6, 3, 7, 18, 17, 20$ סופית: $3, 7, 6, 17, 20, 18, 15$

ניסוח המשפט	שרטוט	כתיב מתמטי
TREE-SEARCH, MINIMUM, MAXIMUM, SUCCESSOR, PREDECESSOR – ירידה (או עלייה) במסלול אחד. **עוקב:** אם יש ימני – המינימום שלו; אחרת עולים עד האב הראשון שאנחנו בתת-העץ השמאלי שלו. (עמ' 216–219)		$O(h)$
TREE-INSERT – יורדים כמו בחיפוש ומוסיפים עלה. **TREE-DELETE** – 0 ילדים: מוחקים; ילד 1: מחברים את הילד לאב; 2 ילדים: מחליפים בעוקב (שאין לו ילד שמאלי). (עמ' 220–222)		$O(h)$, ו-h יכול להיות $n-1$ (עץ "שרשרת")

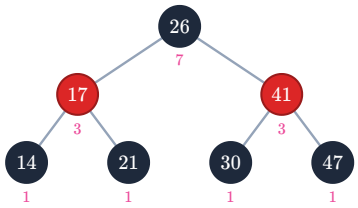
7.2 עץ אדום-שחור

ניסוח המשפט	שרטוט	כתיב מתמטי
5 התכונות: (1) כל צומת אדום או שחור; (2) השורש שחור; (3) כל עלה (NIL) שחור; (4) לאדום – שני ילדים שחורים; (5) מכל צומת, בכל המסלולים לעלים – אותו מספר שחורים ($bh(x)$). (עמ' 229–230)		
למה 13.1 – גובה עץ אדום-שחור עם n צמתים פנימיים הוא לכל היותר $2\lg(n+1)$; ולכן כל פעולות עץ החיפוש ב-$O(\lg n)$. (עמ' 231–232)		$n \geq 2^{bh}-1,\ bh \geq h/2 \Rightarrow h \leq 2\lg(n+1)$
סיבוב – LEFT-ROTATE / RIGHT-ROTATE שומר על תכונת עץ החיפוש, $O(1)$. (עמ' 234–235)		
RB-INSERT – מכניסים כמו TREE-INSERT, צובעים את z **אדום**, ו-RB-INSERT-FIXUP מתקן כל עוד $p[z]$ אדום (y – הדוד): **מקרה 1** – y אדום: $p[z]$ ו-y שחורים, הסב אדום, $z \leftarrow$ הסב (עולים). **מקרה 2** – y שחור ו-z בן ימני: סיבוב שמאלה סביב $p[z]$ – עוברים למקרה 3. **מקרה 3** – y שחור ו-z בן שמאלי: $p[z]$ שחור, הסב אדום, סיבוב ימינה סביב הסב. בסוף – השורש שחור (המקרים המשוקפים – כש-$p[z]$ בן ימני). (עמ' 235–241)		$O(\lg n)$, לכל היותר 2 סיבובים

ניסוח המשפט	שרטוט	כתיב מתמטי
RB-DELETE – מוחקים כמו TREE-DELETE; אם הצומת שהוצא שחור, ל-x (שתפס את מקומו) יש "שחור נוסף", ו-RB-DELETE-FIXUP מתקן (w – האח של x): **מקרה 1** – w אדום: החלפת צבעים וסיבוב סביב $p[x]$ – האח החדש שחור. **מקרה 2** – w שחור ושני בניו שחורים: w אדום, והשחור הנוסף עולה ל-$p[x]$. **מקרה 3** – w שחור, בנו השמאלי אדום והימני שחור: החלפת צבעים וסיבוב סביב w – עוברים למקרה 4. **מקרה 4** – w שחור ובנו הימני אדום: החלפת צבעים וסיבוב סביב $p[x]$ – מסיימים. (עמ' 242–246)		$O(\lg n)$, לכל היותר 3 סיבובים

8 הרחבת מבני נתונים

8.1 עץ ערכי מיקום

ניסוח המשפט	שרטוט	כתיב מתמטי
עץ ערכי מיקום – עץ אדום-שחור עם $size[x]$ (מספר הצמתים בתת-העץ, $size[nil] = 0$): $size[x] = size[left[x]] + size[right[x]] + 1$ – בשרטוט $size$ מתחת לכל צומת. (עמ' 253–254)		
OS-SELECT(x, i) – האיבר ה-i בגודלו: $r = size[left[x]] + 1$; אם $x - i = r$; אם $i < r$ – שמאלה; אחרת ימינה עם $i - r$. (עמ' 254–255)		$O(\lg n)$
OS-RANK(T, x) – הדרגה של x: $r = size[left[x]] + 1$, ועולים לשורש; כל פעם ש-x **בן ימני**, מוסיפים $size[left[p]] + 1$. (עמ' 255–256)		$O(\lg n)$
תחזוקת $size$ – בהכנסה: $+1$ לכל צומת במסלול; בסיבוב – רק שני הצמתים שהסתובבו, בשתי שורות. (עמ' 257)		$size[y] \leftarrow size[x]$ $size[x] \leftarrow size[left[x]] + size[right[x]] + 1$
משפט 14.1 – שדה f שמחושב מ-x ומשני ילדיו בלבד – נשמר בהכנסה ובמחיקה בלי לשנות את הזמן $O(\lg n)$. (עמ' 259)		$f[x] = F(x, f[left[x]], f[right[x]])$

8.2 תכנון מבנה נתונים מורחב

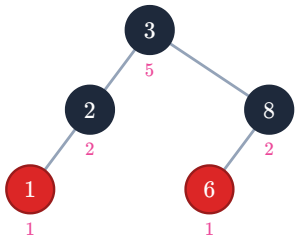
אלגוריתם להרחבת מבנה נתונים (ארבעת השלבים, סעיף 14.2, עמ' 258–259):

1. **בחר מבנה תשתיתי** – בדרך כלל עץ אדום-שחור (הכנסה, מחיקה וחיפוש ב-$O(\lg n)$).
2. **קבע את המידע הנוסף** שיש לתחזק בכל צומת ($size$, max, ...).
3. **ודא שאפשר לתחזק אותו** בהכנסה ובמחיקה (ובסיבובים) בלי לפגוע בזמן – לפי משפט 14.1: מספיק שהשדה מחושב מהצומת ומשני ילדיו.
4. **פתח את הפעולות החדשות** בעזרת המידע הנוסף – כל אחת ירידה או עלייה אחת במסלול: $O(\lg n)$.

לדוגמה נתון (תרגיל 14.1-7, עמ' 258): למנות את מספר ההיפוכים (בעיה 2-4) במערך בגודל n בזמן $O(n \lg n)$ – בעזרת עץ ערכי מיקום.
הדוגמה של בעיה 2-4: $A = \langle 2, 3, 8, 6, 1 \rangle$.
נפעיל את האלגוריתם כדלקמן:

מכניסים את $A[1], A[2], \ldots$ לפי הסדר; אחרי הכנסת $A[j]$: $r = \text{OS-RANK}(T, A[j])$

מבין j האיברים שבעץ, r קטנים או שווים ל-$A[j]$, ולכן $j - r$ **גדולים** ממנו והוכנסו **לפניו** – אלה ההיפוכים שלו



$2 : 1 - 1 = 0 \quad 3 : 2 - 2 = 0 \quad 8 : 3 - 3 = 0$
$6 : 4 - 3 = 1 \quad 1 : 5 - 1 = 4$

נקבל אפוא את המבוקש:

$0 + 0 + 0 + 1 + 4 = 5$ היפוכים; n הכנסות ו-n קריאות ל-OS-RANK, כל אחת $O(\lg n)$: סך הכול $O(n \lg n)$ ($size$ מתחת לכל צומת)

8.3 ערמה מול עץ אדום-שחור – זמני הפעולות

פעולה	ערמת מקסימום	עץ אדום-שחור (עם $size$)
בנייה מ-n איברים	$O(n)$ – BUILD-MAX-HEAP (עמ' 111)	$O(n \lg n)$ – n הכנסות
מקסימום	$\Theta(1)$ – $A[1]$	$O(\lg n)$ – TREE-MAXIMUM
הוצאת המקסימום	$O(\lg n)$	$O(\lg n)$
הכנסה / מחיקה	$O(\lg n)$	$O(\lg n)$
חיפוש מפתח כלשהו	$O(n)$	$O(\lg n)$
עוקב / קודם	$O(n)$	$O(\lg n)$ (עמ' 218)
האיבר ה-i בגודלו / דרגה	–	$O(\lg n)$ – OS-SELECT / OS-RANK (עמ' 254–256)
הדפסה ממוינת	$O(n \lg n)$ – מיון ערמה	$\Theta(n)$ – סריקה תוכית