

מבנה נתונים
ויעילות אלגוריתמים

היקף התכנית: 240 שעות

- עיוני: 192 ש"ש
- מעשי: 48 ש"ש

דרישות קדם

הידע המוקדם הנדרש ללימוד הפרק הזה הוא הכרת מדעי המחשב בהיקף של 5 יחידות לימוד מלימודי התיכון. תלמידים שאין להם ידע מוקדם יצטרכו להשלימו; החומר **לא** יילמד במסגרת קורס זה. כמו כן נדרש ידע מוקדם בשפת C.

אוכלוסיית היעד

כיתות י"ג במגמת הנדסת תוכנה.

מטרות הקורס

- להקנות את עיקרי הגישה המערכתית.
- לפתח את יכולת החשיבה המופשטת באמצעות הגדרת כלים מורכבים יותר מהכלים הבסיסיים שנותנת שפת התכנות, בייחוד באמצעות הגדרת טיפוסים נתונים מופשטים.
- להכיר טיפוסים נתונים מופשטים ידועים (כגון: רשימה, מחסנית, תור, עץ בינרי וגרף) ואת השימוש בהם לפתרון בעיות נתונות.
- להגדיר טיפוסים נתונים מופשטים חדשים וליישם.
- להקנות יכולת לנתח את יעילותם של אלגוריתמים ואת התכניות המיישמות אותם.
- להכיר אלגוריתמים המאפשרים פעולות מתקדמות בטיפוסים נתונים שונים; לעשות בחינה השוואתית של יעילות האלגוריתמים הנלמדים ביחידה.
- להקנות יכולת לבחור טיפוסים נתונים המתאימים ליישום פתרון לבעיה, להגדיר את הטיפוסים, להעמידם לרשות המשתמש באמצעות כתיבת ממשקים מתאימים ויישומם בשפת התכנות הנלמדת.

משרד החינוך
מנהל מדע וטכנולוגיה
מגמת הנדסת תוכנה

טבלת הפרקים וחלוקת השעות המוצעת

פרקי הלימוד	עיוני	מעשי
1 מבוא למבני נתונים	12	2
2 מחסנית	10	4
3 תור	6	2
4 רקורסיה	18	8
5 יעילות אסימפטוטית של אלגוריתמים	20	4
6 רשימות מקושרות	22	6
7 עצים	30	8
8 גרפים	30	4
9 שיטות מיון	18	3
10 שיטות חיפוש	16	3
11 חזרות ומבחנים	10	4
סך הכול	192	48

סביבת העבודה

שפת התכנות האפשרית בקורס לפי תכנית לימודים זו היא C.

פרק 1: מבוא למבני נתונים

- עיוני: 12 שעות
- מעבדה: 2 שעות

מטרות הפרק

- להכיר את הצורך במבני נתונים.
- ללמוד את מבנה הנתונים מערך כאוסף לינארי של משתנים מאותו הסוג.
- להכיר את המושג 'טיפוס נתונים'.
- להכיר את מושג המחלקה הבסיסי.
- לבנות מבנים מורכבים.
- לעשות היכרות מעמיקה עם המחלקה בסביבת העבודה.
- לחדד את רעיון המודולריות של מערכת תוכנה באמצעות בנייה של מחלקות ושימוש בהן.

פירוט התכנים

חזרה – טיפוס נתונים בשפת C

1.1	מערכים ב-C
1.1.1	מערכים חד-ממדיים
1.1.1.1	מימוש מערכים חד-ממדיים
1.1.1.2	מימוש מחרוזות תווים באמצעות מערכים
1.1.1.3	מיזוג מערכים
1.1.2	מערכים דו-ממדיים
1.1.2.1	מימוש מערך דו-ממדי
1.1.2.2	אחסון מטריצות: משולשות, תחתונות, עליונות ורב-אלכסוניות ופעולות עליהן
1.1.3	מערכים רב-ממדיים
1.1.3.1	מימוש מערך רב-ממדי
1.2	מבנים (רשומות) ב-C
1.2.1	הגדרת מבנה והפעולות עליו
1.2.2	מימוש מחרוזות תווים מתוך שימוש במבנים
1.2.3	מימוש מבנים
1.2.4	איגוד (Unions)
1.2.4.1	מימוש Unions

פרק 2: מחסנית

- עיוני: 10 שעות
- מעבדה: 4 שעות

מטרות הפרק

- לתרגל הגדרה של טיפוס נתונים מופשט.
- להכיר טיפוס הנתונים המופשט 'מחסנית' על ידי שימושים שונים.
- לתרגל את מימוש טיפוס הנתונים מחסנית.

פירוט התכנים

2.1	הגדרת המחסנית והפעולות המתבצעות בה
2.2	ייצוג מחסנית ב-C
2.2.1	מימוש מחסנית באמצעות מערך
2.2.2	מימוש הפעולות empty, push, pop, stacktop
2.2.3	מחסנית של טיפוס נתונים שונים
2.2.4	דוגמה: ייצוג תוכי, ייצוג תחילי וייצוג סופי
2.2.4.1	חישוב ביטוי בייצוג סופי (postfix)
2.2.4.2	המרת ביטוי מייצוג תוכי (infix) לייצוג סופי (postfix)
2.2.4.3	חישוב ערך של ביטוי בינרי
2.2.4.4	בניית תמורות בעזרת מחסנית

פרק 3: תור

- עיוני: 6 שעות
- מעבדה: 2 שעות

מטרות הפרק

- לתרגל הגדרה של טיפוס נתונים מופשט.
- לערוך היכרות עם טיפוס הנתונים המופשט 'תור' על ידי שימושים שונים.
- לתרגל את מימוש טיפוס הנתונים תור.

פירוט התכנים

3.1	הגדרת התור והפעולות המתבצעות עליו
3.2	ייצוג תור ב-C
3.3	מימוש תור באמצעות מערך
3.3.1	מימוש פעולות התור
3.3.2	ייצוג תור באמצעות מערך מעגלי
3.3.3	מימוש פעולות התור כאשר התור מיוצג באמצעות מערך מעגלי

פרק 4: רקורסיה

- עיוני: 18 שעות
- מעבדה: 8 שעות

מטרות הפרק

- להכיר את המושג 'רקורסיה' ככלי לפתרון בעיות.
- להכיר את היתרונות והחסרונות של כתיבה רקורסיבית.
- להעמיק בשיטת התכנון מלמעלה למטה.
- לפתח חשיבה רקורסיבית.

פירוט התכנים

4.1	הגדרות ותהליכים רקורסיביים
4.1.1	פונקציות העצרת
4.1.2	כפל של מספרים טבעיים
4.1.3	סדרת פיבונצ'י
4.1.4	חיפוש בינרי
4.1.5	מציאת איבר מינימלי ומקסימלי
4.2	רקורסיה ב-C
4.2.1	פונקציות ב-C ואלגוריתמים רקורסיביים
4.3	פיתוח פתרונות רקורסיביים ותכנותם
4.3.1	חישוב פונקציית עצרת
4.3.2	כפל של מספרים טבעיים
4.3.3	סדרת פיבונצ'י
4.3.4	חיפוש בינרי
4.3.5	מציאת איבר מינימלי ומקסימלי במערך
4.3.6	מציאת מחלק משותף מקסימלי
4.3.7	זיהוי מאפייני רקורסיה
4.3.8	בעיית מגדלי הנוי – הצגת הבעיה ופיתוח הפתרון הכללי
4.3.9	חישוב ערכו של פולינום על משתנה x
4.3.10	המרה מייצוג תחילי לייצוג סופי (postfix) תוך שימוש ברקורסיה
4.3.11	המרה מייצוג תחילי לייצוג בינרי תוך שימוש ברקורסיה
4.3.12	מציאת כל הפרמוטציות האפשריות של סדרת המספרים
4.3.13	מציאת כל הקבוצות החלקיות של קבוצה נתונה
4.3.14	בעיית שמונה המלכות
4.3.15	פתרון בעיות קומבינטוריות באמצעות שימוש ברקורסיה

- 4.3.16 עקרונות של הפרד ומשול (divide & conquer) ודוגמאות:
א. מיון באמצעות מיזוג (merge sort)
ב. חישוב של חזקה
ג. מציאת איבר מינימלי ומקסימלי

פרק 5: יעילות אסימפטוטית של אלגוריתמים

- עיוני: 20 שעות
- מעבדה: 4 שעות

מטרות הפרק

- להבין את המושג 'יעילות', באמצעות הכרת המדד של סיבוכיות זמן הריצה, וכן את חשיבותו של מדד זה.
- ללמוד מושגים בסיסיים בתחום היעילות (המקרה הגרוע ביותר).
- לאבחן שמדד טוב לסיבוכיות הזמן של אלגוריתם הוא מספר הצעדים הבסיסיים שהאלגוריתם מבצע כתלות באורך הקלט.
- להכיר את המושג 'סדר גודל' (O גדול).
- לבצע ניתוח של סיבוכיות זמן ריצה של אלגוריתם.

פירוט התכנים

- 5.0 פונקציות שכיחות ותכונותיהן, התנהגות אסימפטוטית והגדרת המושגים סדר גודל, סיבוכיות זמן ריצה של אלגוריתמים.
- 5.1 מדידת יעילות של אלגוריתם
- 5.1.1 מדידת יעילות
- 5.1.2 שיפור בסדר גודל
- 5.1.3 סדר גודל
- 5.1.4 ניתוח המקרה הגרוע
- 5.1.5 שיפור של אלגוריתם בקבוע
- 5.1.6 סיבוכיות מקום
- 5.2 זמן הריצה של חיפוש בינרי
- 5.3 דוגמאות ופתרונות
- 5.4 סימון אסימפטוטי: Ω , Θ , $\sim$, הגדרות, תכונות ומשפטים
- 5.5 סיבוכיות זמן ריצה מעריכית (אקספוננציאלית)
- 5.6 נוסחאות נסיגה
- 5.7 שיטות בדבר פיתוח נוסחאות נסיגה
- 5.7.1 א. שיטות איטרציה

ב. שימוש בתבניות: שלושה משפטים עיקריים לשיטת איטרציה לנוסחאות נסיגה השייכות לפחות לאחת המשפחות האלה:

1. עבור a, b, c שלמים כך ש: $a \geq 1, b > 1, c > 0$ ו- n הוא חזקה של b ומתקיים:

$$\begin{aligned} T(1) &= c \\ T(n) &= aT(n/b) + cn \end{aligned} \quad d \quad n = 1 \quad .2$$

$$\begin{aligned} T(n) &= aT(n/b) + c \quad n > 1 \\ T(n) &= aT(n/b) + f(n) \end{aligned} \quad \text{משפט אב -} \quad .3$$

.4. ניחוש והוכחה באינדוקציה

5.7.2 פיתוח נוסחאות נסיגה לינאריות הומוגניות ולא-הומוגניות בעזרת משוואת הפרשים

5.7.3 פתרון נוסחאות נסיגה בעזרת החלפת משתנים

5.8 פיתוח אלגוריתמים יעילים ל:

5.8.1 חישוב X^n בזמן $O(\log n)$

5.8.2 בעיית שני מספרים שסכומם x (קבוע נתון) במערך ממורן של n מספרים שלמים וחייביים, בזמן $O(n \log n)$ ובזמן $O(N)$

5.8.3 חישוב האיבר ה- n בסדרת פיבונצ'י בזמן $O(\log n)$ ובזמן $O(n)$

5.8.4 מציאת מחלק משותף מקסימלי

5.8.5 מציאת חציון וערכי מיקום בזמן לינארי – אלגוריתם Select

5.8.6 מציאת Transpose של מטריצה בזמן לינארי

5.8.7 חיפוש ומיון במערכים

5.8.8 התאמת מחרוזות (Pattern matching) בעלי זמן ריצה לא לינארי ובעלי זמן ריצה לינארי

5.8.9 מציאת מכפלה משותפת מינימלית (lcm)

פרק 6: רשימות מקושרות

• עיוני: 22 שעות

• מעבדה: 6 שעות

מטרות הפרק

- להכיר את טיפוס הנתונים 'רשימה'.
- להטמיע את העיקרון של הסתרת מידע באמצעות היכרות עם יישומים שונים לאותו ממשק.
- להכיר את המנגנון של הקצאת זיכרון.

- לממש טיפוס נתונים מופשט באמצעות שימוש בטיפוס נתונים קיים (מחסנית ותור הממומשים בעזרת רשימה).
- להכיר את טיפוס הנתונים המופשט 'רשימה' על ידי שימושים שונים.

פירוט התכנים

- 6.1 רשימה מקושרת לינארית
- 6.2 הכנסה והוצאה של צמתים ברשימה מקושרת לינארית
- 6.3 מימוש מקושר של מחסניות
- 6.4 מימוש מקושר של תורים
- 6.5 החסרונות והיתרונות של הרשימה המקושרת
- 6.6 מימוש הפעולות ברשימות מקושרות
- 6.7 מימוש רשימות ב-C באמצעות מערך
- 6.8 צומתי כותר (header nodes)
- 6.9 רשימות מעגליות
 - 6.9.1 המחסנית כרשימה מעגלית
 - 6.9.2 התור כרשימה מעגלית
 - 6.9.3 פעולות על רשימות מעגליות
 - 6.9.4 בעיית ג'וזפוס (Josephus)
 - 6.9.5 חיבור מספרים חיוביים "מאוד גדולים"
- 6.10 רשימות דו-מקושרות
 - 6.10.1 פעולות על רשימות דו-מקושרות
 - 6.10.2 דוגמה: חיבור מספרים שלמים ארוכים תוך שימוש ברשימות דו-מקושרות
- 6.11 עיבוד רשימות ב-C
 - 6.11.0 מימוש רשימות תוך שימוש באחסון דינמי
 - 6.11.1 מימוש רשימות מקושרות תוך שימוש במשתנים דינמיים
 - 6.11.2 פעולות על רשימות תוך שימוש במימוש הדינמי
 - 6.11.3 השוואת מימושים של רשימות מקושרות מימוש דינמי מול מימוש באמצעות מערך
- 6.12 דוגמה: מטריצות דלילות
 - 6.12.1 ייצוג מטריצה דלילה כרשימה רב-מקושרת
 - 6.12.2 מימוש פעולות על מטריצות דלילות
- 6.13 רקורסיה ברשימות מקושרות
 - דוגמאות: ספירת מסר צמתים ברשימה
 - יצירת רשימה זהה לרשימה נתונה, הפיכת רשימה, מיון רשימה בשיטת מיון באמצעות מיזוג וכולי
- 6.14 טיפול בפולינומים חד/רב-משתנים שרוב מקדמיהם אפס

מבנה נתונים כרשימה מקושרת	6.14.1
מימוש הפעולות: חיבור וכפל של פולינומים	6.14.2
6.15 הדגמת מערכת תורים (סימולציה) בבנק	
ניתוח הבעיה	6.15.1
קביעת מבני נתונים	6.15.2
כתיבת האלגוריתם	6.15.3
כתיבת התכנית	6.15.4

פרק 7: עצים

- עיוני: 30 שעות
- מעבדה: 8 שעות

מטרות הפרק

- להכיר את טיפוס הנתונים 'עץ בינרי' והשימושים השונים שלו.
- לתרגל את השימוש בשגרות רקורסיביות ולהעריך את יעילותן.
- להכיר מהו עץ חיפוש בינרי ואת השימוש בו למיון.
- להכיר מהו עץ כללי ואת השימוש בו ל"עצי משחק".
- להכיר מהו עץ ערמה בינרית ואת השימוש בו למיון.

פירוט התכנים

7.1 רקע כללי	
7.1.1 הגדרת מונחים: צומת פנימי, עלה, דרגה של צומת, תת-עץ שמאלי וימני, רמה של צומת, אב, ילדים (בנים), צאצאים, רמת העץ (גובה-עומק), מרחק פנימי בעץ, מרחק ממוצע בעץ	
7.1.2 הגדרה של עץ בינרי ועץ שלם ברמה n	
7.1.3 הוכחת משפט: אם n מספר צמתים בעץ ו- m דרגה מקסימלית של צומת בעץ, אז עומק העץ (H) הוא: $H > \log_m [n(m-1)+1]-1$	
7.1.4 חישוב גובה בעץ בינרי	
7.1.5 הוכחת משפט עיקרי בעצים בינריים:	
מספר העלים בעץ בינרי גדול ב-1 ממספר הצמתים בעלי דרגה 2	
7.1.6 מציאת עצים בינריים באמצעות n צמתים	
הערות: להלן הגדרות של מונחים שלא מופיעים בספרי לימוד –	
1. דרגה של צומת היא מספר תת-עצים היוצאים ממנו	
2. מרחק פנימי בעץ הוא סכומי המרחקים מכל צומת לשורש העץ	
3. מרחק ממוצע בעץ הוא מרחק פנימי בעץ מחולק למספר הצמתים בעץ	
7.2 עצים בינריים	
7.2.1 "יצוג עץ בינרי"	

משרד החינוך
מנהל מדע וטכנולוגיה
מגמת הנדסת תוכנה

יישומים של עצים בינריים למיון רשימת מספרים	7.2.2
מימוש עץ בינרי באמצעות מערך ובאמצעות מבנה דינמי	7.2.3
סריקת עץ בינרי באמצעות סריקה בסדר תחילי (preorder), תוכי (inorder) וסופי (postorder) – בצורה רקורסיבית ואיטרטיבית	7.2.4
עצים בינריים הטרוגניים	7.2.5
עץ בינרי לחלוטין (למהדרין) (strictly binary tree)	7.3
יישום עצים בינריים להצפנת הודעות – אלגוריתם הופמן	7.3.1
עצים כמעט שלמים (almost complete binary)	7.4
יישום עצים שלמים לפתרון מחודש של בעיית ג'וזפוס (Josephus)	7.4.1
עצים כלליים	7.5
הגדרת עצים כלליים וייצוגם ב-C	7.5.1
סריקת עצים כלליים	7.5.2
פעולות בעצים כלליים	7.5.3
רשות דוגמה: עצי משחק (משחק טיק-טק-טו המכונה גם איקס-מיקס-דריקס), או משחק MIN או משחק אחר	7.5.4
הצגת הבעיה וניתוחה	7.5.4.1
קביעת מבנה הנתונים והאלגוריתם	7.5.4.2
עיקרון מינימקס	7.5.4.3
כתיבת התכנית	7.5.4.4
ערמה (heap)	7.6
ערמת מקסימום, הגדרה ומימוש ערמה בעזרת עץ בינרי, הממומש על ידי מערך חד-ממדי	7.6.1
אלגוריתם Heapify	7.6.2
תכונות הערמה	7.6.3
הוספת איבר חדש וביטול איבר מינימלי בערמת מקסימום בזמן $O(\log n)$	7.6.4
בניית ערמה מסדרה אקראית בעלת n איברים בזמן $O(n)$	7.6.5
מיון ערמה	7.6.6
ערמת מינימום	7.6.7
ייצוג קבוצות ופעולות עליהן באמצעות עצים בזמן $o(n \log n)$ (tree structure for the union-find problem)	7.7

פרק 8: גרפים

- עיוני: 30 שעות
- מעבדה: 4 שעות

מטרות הפרק

- להכיר את טיפוס הנתונים 'גרף' ואת השימושים השונים שלו.
- לתרגל את השימוש בשגרות רקורסיביות ולהעריך את יעילותן.
- להכיר מהו מסלול קצר ביותר ואת השימוש בו.
- להכיר מהי זרימה מקסימלית ואת השימוש בו.
- להכיר מהו עץ פורש מינימלי ואת השימוש בו.

פירוט התכנים

- 8.1 הגדרת הגרף (גרף לא-מכוון, מכוון, מסלול, קודקוד סמוך, גרף פשוט, מולטי-גרף, מסלול, מעגלי, גרף מעגלי, גרף שלם, גרף קשיר, גרף קשיר מכוון בחוזקה, גרף דו צדדי ומלא).
- 8.2 פעולות בסיסיות על גרף
- 8.3 יישום של גרפים – מסלול בין שתי ערים, רשתות מים, כבישים, מסלולי טיסה
- 8.4 ייצוג עקיב של גרף ב-C ומימוש הפעולות הבסיסיות על גרף
- 8.5 סגור טרנזיטיבי – מטריצת מסלולים באמצעות שתי שיטות:
 - א. כפל מטריצות (אלגוריתם בעל סיבוכיות $O(n^4)$, כאשר n מציין מספר קודקודים בגרף)
 - ב. אלגוריתם Floyd-Warshall (אלגוריתם בעל סיבוכיות $O(n^3)$)
- 8.6 דוגמאות ליישומים של גרפים:
 - 8.6.1 בעיית זרימה ואלגוריתם של Ford Fulkerson – לזרימה אופטימלית
 - 8.6.2 סריקת גרף באמצעות סריקה
 - א. לעומק (Dfs) ויעילותו
 - ב. לרוחב (Bfs) ויעילותו
 - ג. רכיבי קשירות בגרף לא-מכוון
 - ד. רק"ח (רכיבי קשירות בחוזקה) בגרף מכוון
 - ה. מציאת רק"ח בזמן לינארי
 - ו. גרף-על ותכונותיו
 - 8.6.3 מיון טופולוגי ומסלולים קריטיים ברשת (Pert) בזמן לינארי
 - 8.6.4 מציאת מסלולים מינימליים מכל צומת לכל צומת בגרף לפי:
 - א. דיקסטרה

- ב. בלמן פורד
ג. פלויד-וורשל
ד. DAG – מסלול מינימלי בגרף מכוון חסר מעגלים (ייתכן גם עם משקולות שליליים) בזמן לינארי
ה. SPIRA (רשות)
8.6.5 עץ פורש מינימלי
8.6.5.1 אלגוריתם גרידי
8.6.5.2 אלגוריתם kruskal
8.6.5.3 אלגוריתם prim
8.6.6 חקירת יעילותם של מעגלים אופטימליים בגרף:
8.6.6.1 מסלול ומעגל אוילרי
8.6.6.2 אלגוריתם למציאת מעגל, מסלול אוילרי
8.6.6.3 מסלול ומעגל המילטוני
8.7 ייצוג מקושר של גרפים
8.7.1 ייצוג גרף באמצעות מבנה רב-מקושר
8.7.2 מימוש הפעולות הבסיסיות בגרף בייצוג המקושר
8.7.3 יישום של גרף מקושר – בעיית תזמון
8.8 אלגוריתמים מתקדמים בגרפים (רשות)

פרק 9: מיון

- עיוני: 18 שעות
- מעבדה: 3 שעות

מטרות הפרק

- להכיר את בעיות המיון והמיזוג ואלגוריתמים שונים לפתרונן.
- לתרגל ולהעמיק בנושאים הנלמדים ביחידה.

פירוט התכנים

- 9.1 סיווג מיומנים
9.2 כדאיות המיון ושיקולי יעילות
9.3 מיוני החלפה
9.3.1 מיון בועות (bubble sort)
9.3.2 מיון מהיר (quicksort)
9.4 מיוני בחירה
9.4.1 מיון בחירה ישירה
9.4.2 מיון בחירה באמצעות עצים בינריים

מיון טורניר (tournament)	9.4.3
מיון ערמה (heapsort)	9.4.4
מיוני הכנסה	9.5
מיון הכנסה פשוטה	9.5.1
מיון shell	9.5.2
מיוני הפצה	9.6
מיון בסיס (radixsort)	9.6.1
מיון בזמן לינארי	9.6.2

פרק 10: חיפוש

- עיוני: 16 שעות
- מעבדה: 3 שעות

מטרות הפרק

- להכיר את בעיות החיפוש הבסיסי והמתקדם ואלגוריתמים שונים לפתרונן.
- לתרגל ולהעמיק בנושאים הנלמדים ביחידה.
- להכיר מהו עץ חיפוש בינרי מאוזן ואת השימוש בו למיון.
- להכיר מבנה נתונים שכיח – טבלת HASH.

פירוט התכנים

שיטות חיפוש בסיסיות	10.1
מונחים בסיסיים	10.1.1
חיפוש סדרתי	10.1.2
חיפוש סדרתי בטבלה מסודרת	10.1.3
חיפוש אינדקס סדרתי	10.1.4
חיפוש בינרי	10.1.5
חיפוש באמצעות עץ חיפוש בינרי	10.2
עצים בינריים מאוזנים (עצי avl)	10.3
המרת מפתחות (hashing)	10.4
פונקציות המרה	10.4.1
פתרון התנגשות המרה על ידי המרה חוזרת	10.4.2
פתרון התנגשות המרה על ידי שרשור	10.4.3
דוגמה: אלגוריתם הופמן	10.4.4

10.5 עצי 2-3
פעולות הכנסה וביטול אלמנט בעץ 2-3
10.6 10.6 עצי B ו/או B+.

ביבליוגרפיה

ספרי לימוד

1. חוטובלי, ראובן וחוטובלי, פנינה, **רקורסיה (מבני נתונים חלק א)**. הוצאת אורט ישראל, 2002.
2. חוטובלי, ראובן וחוטובלי, פנינה, **יעילות של אלגוריתמים (מבני נתונים חלק ב)**. הוצאת אורט ישראל, 2003.
3. Horowitz, E., Mehta, D., Sahni, S., *Fundamentals of Data Structures in C++*. New York: Freeman, 1993.

ספרות עזר

1. Corman, Thomas H., Leiserson, Charles Eric, Rivest, Ronald L., *Introduction to Algorithms*. McGraw Hill, 2001.
2. Augenstein, M., Langsam, Y., Tenenbaum, A., *Data Structures Using C and C++*. Prentice-Hall, 1996.