



מכללת אורט כפר-סבא

## מבני נתונים ויעילות אלגוריתמים

### תרגיל מס' 4

פתרו את השאלות הבאות. יש לסיים את התרגיל עד יום א' (19.10).

#### שאלה 1

א. ממשו את הפונקציה שכותרתה:

```
void swap (int *a, int *b)
```

הפונקציה תקבל מצביעים לשני משתנים מטיפוס שלם, ותבצע החלפה ביניהם.

ב. ממשו את הפונקציה שכותרתה:

```
void bubble_sort (int a[], int n)
```

הפונקציה ממיינת בסדר עולה מערך a המכיל n מספרים שלמים, בעזרת האלגוריתם

למיון בועות (Bubble Sort). היעזרו בפונקציה שכתבתם בסעיף א'.

ג. איזה שינוי עליכם לעשות בפונקציה המממשת את האלגוריתם למיון בועות, אותה

כתבתם בסעיף ב', על מנת שהיא תמיינ את המערך בסדר יורד, במקום בסדר עולה?

#### שאלה 2

בשיעור ראינו כי הן מיון בועות והן מיון בחירה, הם שניהם מסיבוכיות זמן ריצה  $\Theta(n^2)$  במקרה הגרוע ביותר. בשאלה זו ננסה להעריך את סיבוכיות זמן הריצה של שני אלגוריתמים אלו במקרה הטוב ביותר. כזכור, בשיעור ראינו כי באלגוריתם למיון בועות ניתן להכניס שיפור, בדמות הדגל הבוליאני sorted. הניחו שזו הגרסה של מיון בועות בה אנו משתמשים.

א. כמה פעולות יבצע האלגוריתם למיון בועות (המשופר) אם הוא מקבל בתור קלט מערך

שהוא כבר ממוין? בטאו את תשובתכם בעזרת הסימן האסימפטוטי  $\Theta$ .

ב. האם קיים קלט אחר מזה שתואר בסעיף א', שעבורו האלגוריתם למיון בועות יבצע פחות פעולות?

ג. בהסתמך על סעיפים א' ו-ב', קבעו מהי סיבוכיות זמן הריצה של האלגוריתם למיון בועות, במקרה הטוב ביותר.

ד. כעת, קבעו מהי סיבוכיות זמן הריצה של האלגוריתם למיון בחירה, במקרה הטוב ביותר. לשם כך, יהיה עליכם קודם לקבוע מה יהיה הקלט הטוב ביותר עבור האלגוריתם.

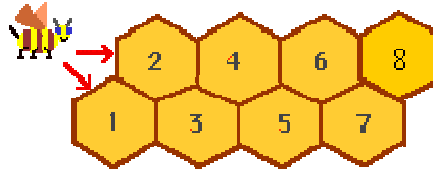
ה. נניח שברשותכם מערך שהוא כמעט ממוין (כלומר: הרוב הגדול של איבריו ממוינים, פרט למספר קטן של איברים שלא נמצאים במקומם). באיזה מבין שני האלגוריתמים למיון שהכרנו – מיון בחירה ומיון בועות – תעדיפו להשתמש?





## שאלה 9

בכמה דרכים שונות יכולה דבורה לעוף לתא מסוים, אם היא מתחילה משמאל לימין (החל מתאים 1 או 2)? נתון שהדבורה מתקדמת מתא אל תא סמוך, כך שתמיד היא עפה מתא אל תא אחר שמספרו גדול ממנו (לדוגמה דבורה יכול לעוף מתא 2 לתא 3 אך לא להיפך).



לדבורה יש רק דרך אחת להגיע לתא מס' 1.

לדבורה יש שתי דרכים להגיע לתא מס' 2: דרך התאים 12, או דרך תא מס' 2.

לדבורה יש שלוש דרכים להגיע לתא מס' 3: דרך 13, או 23, או 123.

בכמה דרכים יכולה הדבורה לעוף כדי להגיע לתא מס' 4? מס' 7? מס' 8?

כתבו פונקציה המקבלת כפרמטר מספר טבעי  $n$ , ומחזירה את מספר הדרכים השונות בהן הדבורה יכולה לעוף כדי להגיע לתא מספר  $n$ .

## שאלה 10

קבוצת תלמידים המכילה בנים  $\text{😊}$  ובנות  $\text{😜}$  משתתפים באירוע בבית ספר. המארגנים רוצים להושיב חלק מהמשתתפים על שורה של כסאות כך ששני בנים לא ישבו בסמוך זה לזה. השאלה היא בכמה אפשרויות ניתן לסדר ב- $n$  כסאות קבוצה של בנים ובנות בשורה כך ששני בנים לא ישבו בסמוך זה לזה.

לדוגמה:

בכסא אחד ניתן להושיב בן  $\text{😊}$  או בת  $\text{😜}$  כלומר 2 אפשרויות

בשני כסאות יש 3 אפשרויות: בן ובת  $\text{😊😜}$  או בת ובן  $\text{😜😊}$  או שתי בנות  $\text{😜😜}$

בשלוש כסאות יש 5 אפשרויות:  $\text{😊😊😊}$ ,  $\text{😊😊😜}$ ,  $\text{😊😜😊}$ ,  $\text{😜😊😊}$ ,  $\text{😜😜😜}$ .

א. בכמה אפשרויות ניתן לסדר בנים ובנות על 4 כסאות, כך ששני בנים לא ישבו זה ליד זה?

ב. בכמה אפשרויות ניתן לסדר בנים ובנות על 5 כסאות, כך ששני בנים לא ישבו זה ליד זה?

ג. כתבו פונקציה המקבלת כקלט מס' טבעי  $n$ , ומחזירה את מס' האפשרויות לסדר בנים

ובנות על  $n$  כסאות, כך ששני בנים לא ישבו זה ליד זה.

**הדרכה:** בנו נוסחה רקורסיבית  $f(n)$  בדומה למה שעשינו בשיעור בבעיית בניית קו

הקרשים. עיינו במצגת השיעור האחרון.

ד. **(סעיף רשות)** כתבו פונקציה המקבלת כקלט מס' טבעי  $n$ , ומחזירה את מס' האפשרויות לסדר

בנים ובנות על  $n$  כסאות, כך ששלושה בנים לא ישבו זה ליד זה (אך שניים כן יוכלו לשבת!).

## שאלה 11

בתרגיל קודם נתקלנו במושג של סדרה הנדסית (שאלה 10 מתוך תרגיל מס' 1 בתכנות מערכות בשפת C). כזכור, מדובר בסדרה שבה קיים יחס קבוע  $q$  בין כל שני איברים סמוכים:

$$a, aq, aq^2, aq^3, aq^4, aq^5, \dots$$

עליכם לכתוב פונקציה רקורסיבית שותרתה:

```
int geometric_sequence (int a, int q, int n)
```

המקבלת כפרמטר שלושה מספר טבעיים  $a, q, n$ , ומחזירה את האיבר ה- $n$ -י של הסדרה ההנדסית שאיברה הראשון הוא  $a$  והמנה הקבועה שלה היא  $q$ .

**הזרקה:** חשבו - מה יהיה תנאי העצירה? חשבו גם על הצעד הרקורסיבי - כיצד ניתן לבטא איבר בסדרה הנדסית על-ידי האיבר הקודם לו?



## שאלה 12

כתבו תכנית מחשב בשפת C המיישמת את האלגוריתם הבא:

קלוט מספר טבעי  $X$

כל עוד  $X$  איננו פלינדרום, בצא:

השם  $Y$  את המספר ההפוך בסדר ספרותיו  $X-f$

השם  $X$  את  $X + Y$

הצא כפלט את צרכו  $fe$   $X$

לדוגמא, אם נפעיל את האלגוריתם על  $X = 87$ , אז נקבל:

$$87 + 78 = 165$$

$$165 + 561 = 726$$

$$726 + 627 = 1353$$

$$1353 + 3531 = 4884 \text{ ונסיים.}$$

אם נפעיל את האלגוריתם על  $X = 64$ , אז נקבל:

$$64 + 46 = 110$$

$$110 + 011 = 121 \text{ ונסיים.}$$

אם נפעיל את האלגוריתם על  $X = 13$ , אז נקבל:  
 $13 + 31 = 44$ , ונסיים.

האם תמיד מובטח שהאלגוריתם יעצור? כלומר – האם נגיע בהכרח בשלב מסוים לפלינדרום?  
מסתבר שמדובר בבעיה פתוחה, שהתשובה עליה לא ידועה. קראו, למשל, את סיפורו (העגום?) של מתכנת שהריץ תכנית זו במחשב במשך **שלוש שנים**, מבלי שהתכנית תיעצר:

<http://www.fourmilab.ch/documents/threeyears/threeyears.html>

**רמז:** האם כתבנו באחד התרגילים בתכנות מערכות בשפת C פונקציה שעשויה להיות שימושית בשאלה זו?

**ולשם שעשוע:** הקומיקאי האמריקאי Weird Al Yankovic הקליט שיר בשם 'Bob', שהוא פארודיה על סגנון השירה של הזמר בוב דילן, אך מה שמיוחד בשיר הוא שכל שורה בו היא פלינדרום: <https://www.youtube.com/watch?v=UnJdxUwF1Wg>



### שאלה 13

מעוניינים לכתוב פונקציה רקורסיבית שכותרתה -

```
void find_min_max (int a[], int *min, int *max, int n)
```

הפונקציה מקבלת מערך  $a$  של מס' שלמים שגודלו  $n$ , ומחזירה (דרך הפרמטרים  $min$  ו- $max$ , המועברים לפי כתובת) את האיבר המינימלי והאיבר המקסימלי במערך.

למשל, נניח ש- $a$  הוא המערך הבא, שגודלו 8 תאים:

0	1	2	3	4	5	6	7
11	0	-8	44	0	16	-3	12

אם נזמן את הפונקציה על-ידי `find_min_max(a, &x, &y, 8)`, כאשר  $x$  ו- $y$  הם משתנים מטיפוס שלם, אזי הם יקבלו את הערכים  $x = -8$ ,  $y = 44$ .

להלן מימוש חלקי של הפונקציה, אשר חסרים בו עשרה ביטויים הממוספרים (1) - (10):

```

void find_min_max (int a[], int *min, int *max, int n)
{
    int temp_min, temp_max;

    if (___ (1) ___)
    {
        ___ (2) ___;
        ___ (3) ___;
    }
    else
    {
        find_min_max(a, ___ (4) ___, ___ (5) ___, ___ (6) ___);
        if (___ (7) ___)
            *min = temp_min;
        else
            ___ (8) ___;

        if (___ (9) ___)
            *max = temp_max;
        else
            ___ (10) ___;
    }
}

```

**הדרכה:** עבדו לפי עיקרון הדומה לזה של הפונקציה max\_array שראינו בשיעור האחרון.

