



מכללת אורט כפר-סבא

מבני נתונים ויעילות אלגוריתמים

תרגיל מס' 5

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

שאלה 1

לפניכם הפונקציה הרקורסיבית הבאה, הקרויה פונקצית אקרמן, על שם ממציאה, המתמטיקאי הגרמני וילהלם אקרמן (Wilhelm Ackermann, 1896-1962):

$$A(m, n) = \begin{cases} n+1 & , m = 0 \\ A(m-1, 1) & , n = 0 \text{ וגם } m > 0 \\ A(m-1, A(m, n-1)) & , n > 0 \text{ וגם } m > 0 \end{cases}$$

- ממשו את פונקצית אקרמן בשפת C.
- הציגו כפלט, בצורה מעוצבת כטבלה (השתמשו ב-\texttt{t}), את כל ערכי $A(m, n)$ עבור m בין 0 ל-3, ו- n בין 0 ל-4.
- הציגו כפלט את $A(4, 0)$ ו- $A(4, 1)$ (התאזרו בסבלנות!).
- נסו לחשב את $A(4, 2)$. התכנית לא תעצור בזמן סביר, שכן הערך המבוקש הוא $2^{65536}-3$. מספר זה גדול יותר ממספר החלקיקים ביקום.

שאלה 2

א. בכל אחד מהסעיפים הבאים מופיע נוסחה נסיגה (רקורסיבית). פתרו אותה באמצעות שיטת האיטרציה (iteration method):

$$(i) \quad T(0) = 0, \text{ כאשר } T(n) = T(n-1) + 1$$

$$(ii) \quad T(1) = 0, \text{ כאשר } T(n) = T\left(\frac{n}{2}\right) + 1$$

$$(iii) \quad T(0) = 0, \text{ כאשר } T(n) = T(n-1) + n$$

ב. (רשות) פתרו את נוסחאות הנסיגה מסעיף א' בשיטת ההצבה (substitution method).

שאלה 3

בשיעור הכרנו את המתמטיקאי הצרפתי Édouard Lucas (1842-1891) שהמציא את חידת מגדלי האנוי. תוצאה מפורסמת נוספת, הקרויה על שמו של לוקאס, היא סדרת מספרים הנראית כך: $2, 1, 3, 4, 7, 11, 18, \dots$

קל לראות שסדרת לוקאס מקיימת את החוקיות הבאה: כל איבר בסדרת לוקאס שווה לסכום שני האיברים שקודמים לו. כבר הכרנו סדרה המקיימת חוקיות כזו, וקראנו לסדרה מפורסמת זו בשם סדרת פיבונצ'י: $1, 1, 2, 3, 5, 8, 13, \dots$

ההבדל בין סדרת לוקאס לסדרת פיבונצ'י הוא זוג האיברים ההתחלתיים (סדרת לוקאס מתחילה ב-1 ו-1, בשעה שסדרת פיבונצ'י מתחילה ב-1 ו-1).

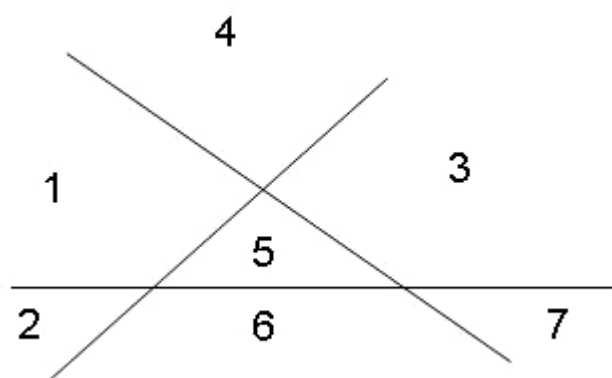
א. כתבו פונקציה רקורסיבית בשפת C המקבלת כפרמטר מספר טבעי n , ומחזירה את האיבר שנמצא במקום ה- n של סדרת לוקאס.

ב. (רשות) נסמן את האיבר שנמצא במקום ה- n של סדרת פיבונצ'י ב- F_n , ואת האיבר במקום ה- n של סדרת לוקאס ב- L_n . הוכיחו (בעזרת אינדוקציה, או בכל דרך אחרת) כי עבור כל $n \geq 3$ טבעי מתקיימת הזהות: $L_n = F_n + F_{n-2}$.

שאלה 4

בשיעור פתרנו את הבעיה של מציאת מספר החלקים הגדול ביותר שניתן לחלק מישור על-ידי n ישרים. ניתן לנסח בעיה זו גם באופן הבא: נתונה עוגה עגולה. פורסים אותה בסכין n פעמים, כאשר בכל פעם חותכים בקו ישר. מהו מס' הפרוסות המקסימלי שניתן לקבל? בסעיפים הבאים נפתור וריאציות שונות על בעיה זו. הסעיפים אינם תלויים זה בזה. סעיפים א', ג' ו-ד' הם סעיפי רשות, וסעיף ב' הוא סעיף חובה.

א. (רשות) כאשר מחלקים את המישור על-ידי n קווים ישרים, חלק מהתחומים הם תחומים סגורים, וחלק מהתחומים הם תחומים פתוחים. נביט, לדוגמא, על האזור הבא:



באזור זה, תחומים 1,2,3,4,6,7 הם פתוחים, ורק תחום 5 הוא סגור. מהו מספר התחומים הפתוחים המתקבלים כאשר מחלקים את המישור על-ידי n קווים ישרים? הוכיחו את תשובתכם.

- ב. כתבו פונקציה רקורסיבית בשפת C אשר מקבלת כפרמטר מספר טבעי (שלם וחיובי) n , ומחזירה את מספר האזורים המקסימלי הנוצרים במישור על-ידי n מעגלים. ניתן להיעזר בפתרון שעשינו בכיתה לבעיית הקווים הישרים במישור.
- ג. רשות כתבו פונקציה שאינה רקורסיבית הפותרת את אותה הבעיה כמו הפונקציה שכתבתם בסעיף ב'. הוכיחו ששתיהן פותרות את אותה הבעיה.
- ד. רשות מהו מספר החלקים הגדול ביותר שניתן לחלק את המרחב התלת-מימדי על-ידי n מישורים? סמנו מספר זה ב- P_n , חשבו את ערכו עבור מקרים פרטיים קטנים, ונסו להכליל ולהגיע לנוסחת נסיגה. מצאו ביטוי סגור הפותר את נוסחת הנסיגה, והוכיחו שביטוי סגור זה הוא אכן התשובה הנכונה.

שאלה 5

ממשו את הפונקציה שכותרתה:

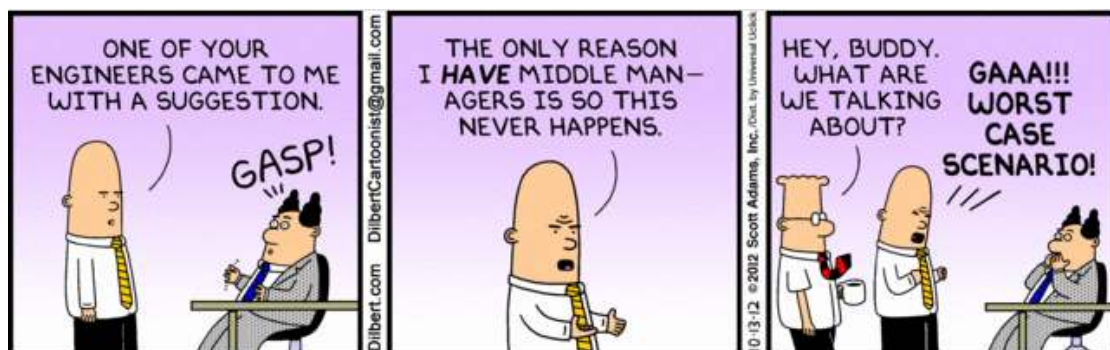
```
void insertion_sort (int a[], int n)
```

הפונקציה ממיינת בסדר עולה מערך a המכיל n מספרים שלמים, בעזרת האלגוריתם למיון הכנסה (Insertion Sort) שלמדנו בכיתה.

שאלה 6

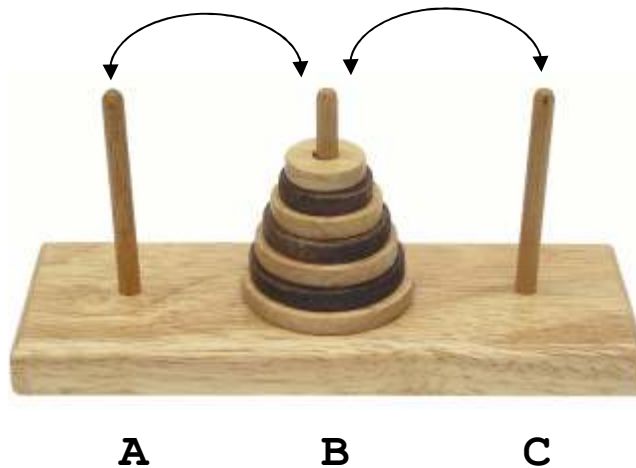
בכל אחד מסעיפי שאלה זו תתבקשו לכתוב גירסה מעט שונה של האלגוריתם למיון הכנסה (Insertion Sort) שראינו בכיתה. שימו לב שאתם נדרשים לכתוב אלגוריתם מילולי, ולא תכנית מחשב. בנוסף, שימו לב שהסעיפים הם בלתי תלויים האחד בשני.

- א. כתבו מחדש את האלגוריתם למיון הכנסה כך שימייין את המערך A בסדר יורד, במקום בסדר עולה.
- ב. כתבו מחדש את האלגוריתם למיון הכנסה כך שימייין את המערך A בסדר עולה, בצורה כזאת שאחרי כל מעבר בלולאה החיצונית, נקבל תת-מערך ממוין בחלק הימני של A .
- ג. כתבו מחדש את האלגוריתם למיון הכנסה כך שימייין את המערך A בסדר יורד, בצורה כזאת שאחרי כל מעבר בלולאה החיצונית, נקבל תת-מערך ממוין בחלק הימני של A .



שאלה 7

בשאלה זו עליכם לכתוב פונקציה המדפיסה הוראות לפתרון בעיית מגדלי האנוי (בדומה לפונקציה hanoi שכתבנו בשיעור), אך בהינתן אילוץ נוסף: בכל צעד, מותר להזיז דיסקה רק בין שני מגדלים סמוכים. כלומר: לא ניתן, על-ידי צעד בודד, להעביר דיסקה ממגדל A למגדל C, או ממגדל C למגדל A. משני מגדלים אלו ניתן להעביר דיסקה רק למגדל B.



לפניכם הגדרת טיפוס:

```
typedef enum {A,B,C} tower;
```

כמו כן, נתונה הפונקציה `constrained_hanoi`, הפותרת את הבעיה. בפונקציה חסרים שישה ביטויים, הממוספרים (1)-(6), ויש להשלים אותם:

```
void constrained_hanoi (int n, tower from, tower to)
{
    tower temp = (A + B + C) - to - from;

    if (n == 0) return;

    if (temp != B) /* סימן ש-to ו-from הם שני מגדלים סמוכים */
    {
        _____ (1) _____;
        printf("move disc from tower %c to tower %c\n", 'A'+from, 'A'+to);
        _____ (2) _____;
    }
    else /* סימן ש-to ו-from הם שני המגדלים שבקצוות */
    {
        _____ (3) _____;
        _____ (4) _____;
        _____ (5) _____;
        printf("move disc from tower %c to tower %c\n", 'A'+temp, 'A'+to);
        _____ (6) _____;
    }
}
```

שאלה 8

כתבו פונקציה בשפת C אשר כותרתה היא :

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

- א. הפונקציה תקבל מערך a של מס' שלמים שגודלו n, ותמיין אותו לפי האלגוריתם למיון בועות **אך בצורה רקורסיבית**: ראשית הפונקציה תסרוק את המערך, תשווה כל שני איברים סמוכים, ותחליף ביניהם אם הם מופיעים בסדר הפוך. בסיום סריקה זו, האיבר הגדול ביותר במערך "ביעבעע" לקצהו הימני. כעת, הפונקציה תמיין באופן רקורסיבי את n-1 האיברים הראשונים של המערך.
- ב. כתבו נוסחת נסיגה $T(n)$ המתארת את זמן הריצה של הפונקציה מסעיף א', ופתרו אותה באמצעות שיטת האיטרציה.
- ג. האם הגרסה הרקורסיבית של מיון בועות יעילה יותר אסימפטוטית (בסדר גודל) מאשר הגרסה האיטרטיבית של האלגוריתם?

שאלה 9

כתבו פונקציה בשפת C אשר כותרתה היא :

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

- א. הפונקציה תקבל מערך a של מס' שלמים שגודלו n, ותמיין אותו לפי האלגוריתם למיון הכנסה **אך בצורה רקורסיבית**: ראשית הפונקציה תמיין באופן רקורסיבי את n-1 האיברים הראשונים של המערך, ולאחר מכן תכניס את האיבר ה-n-י למקום המתאים.
- ב. כתבו נוסחת נסיגה $T(n)$ המתארת את זמן הריצה של הפונקציה מסעיף א', ופתרו אותה באמצעות שיטת האיטרציה.
- ג. האם הגרסה הרקורסיבית של מיון הכנסה יעילה יותר אסימפטוטית (בסדר גודל) מאשר הגרסה האיטרטיבית של האלגוריתם?

שאלה 10 (רשות)

- א. כתבו תכנית מחשב **יעילה** בשפת C המחשבת ומציגה כפלט את סכום כל המספרים הזוגיים בסדרת פיבונצ'י, שערכם אינו עולה על ארבע מיליון. כזכור, סדרת פיבונצ'י היא הסדרה הבאה: ... 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... קל לראות כי המספרים הזוגיים בסדרה הם: ... 2, 8, 34, ... וכו'. הקפידו שהתכנית תהיה **יעילה ככל האפשר**.
- ב. פתרו את הבעיה **ללא שימוש במחשב** (ניתן להשתמש במחשבון). הדרכה: זכרו שהיחס

$$\phi = \frac{1 + \sqrt{5}}{2} = 1.618\dots$$

בין כל שני מספרים עוקבים בסדרת פיבונצ'י הוא בקירוב:

שאלה 11 (רשות)

- א. אם נרשום את כל המספרים הטבעיים הקטנים מ-10 שהם כפולות של 3 או 5, נקבל את המספרים: 3, 5, 6, 9. סכומם של מספרים אלו הוא 23. כתבו תכנית מחשב **יעילה** בשפת C המחשבת ומציגה כפלט את סכום כל הכפולות של 3 או 5 הקטנות מ-1000.
- ב. פתרו את הבעיה **ללא שימוש במחשב** (אין צורך גם להשתמש במחשבון).

בשתי השאלות הבאות נתוודע לאלגוריתם הקרוי **האלגוריתם של אוקלידס** ולפונקציה הקרויה **הפונקציה של אוילר**. הם קיבלו את שמם בזכות המתמטיקאים שגילו אותם, הגיאומטריקאי היווני הקדום אוקלידס (Euclid; 325-265 לפנה"ס) והמתמטיקאי השוויצרי **ליאונרד אוילר** (Leonard Euler; 1707-1783). הם שניהם אנשים חשובים: נסו למצוא עליהם מידע בויקיפדיה...

שאלה 12

האלגוריתם הבא הוא אלגוריתם רקורסיבי המקבל כקלט שני מספרים טבעיים (שלמים וחיוביים) ומחזיר את **המחלק המשותף הגדול ביותר** (ממג"ב) שלהם. הממג"ב (באנגלית: greatest common divisor, או בקיצור gcd) הוא המספר הטבעי הגדול ביותר המחלק את שני המספרים. למשל, $\text{gcd}(12,18)$ שווה ל-6 (מפני ש-6 הוא המספר הגדול ביותר שמחלק את 12 ו-18; גם 1, 2 ו-3 מחלקים את 12 ו-18, אבל 6 גדול יותר מהם).

- א. חשבו את $\text{gcd}(12,8)$ ואת $\text{gcd}(121,44)$.
- ב. שני מספרים נקראים **זרים זה לזה** (co-prime) אם הממג"ב שלהם שווה ל-1. תנו דוגמא לשני מספרים כאלו.

האלגוריתם הבא היה ידוע כבר ליוונים הקדמונים, ומקובל לייחס אותו לאוקלידס:

```
int gcd (int a, int b)
{
  if (b == 0)
    return a;
  else
    return gcd (b, a%b);
}
```



ג. בדקו ידנית בעזרת טבלת מעקב את ריצת האלגוריתם עבור הדוגמאות מסעיפים א' ו-ב'.
 לפניכם גירסה איטרטיבית (לא רקורסיבית; המשתמשת בלולאה) של האלגוריתם של
 אוקלידס. חסרים בה ארבעה ביטויים, הממוספרים (1) עד (4). השלימו אותם. הקפידו
 לבדוק את תשובתכם, על-ידי הרצת הפונקציה עבור דוגמאות קלט מגוונות!

```
int gcd (int a, int b)
{
    int c;
    while ( _____ (1) _____ ) {
        c = _____ (2) _____;
        _____ (3) _____;
        a = _____ (4) _____;
    }
    return a;
}
```

שאלה 13

פונקציית פי (phi) של אוילר $\varphi(n)$ מוגדרת להיות כמות המספרים הקטנים מ- n וזרים ל- n (שני מספרים a ו- b יקראו זרים, אם המחלק המשותף הגדול ביותר שלהם הוא 1). למשל, $\varphi(8) = 4$, מפני שיש ארבעה מספרים הקטנים מ-8 וזרים לו (מספרים אלו הם 1,3,5,7).

א. חשבו את $\varphi(7)$ ואת $\varphi(10)$.

ב. נתונה הפונקציה הבאה, המקבלת כפרמטר מספר שלם וחיובי (טבעי) n ומחזירה את $\varphi(n)$. בפונקציה חסרים ארבעה ביטויים, המסומנים (1) עד (4). השלימו אותם.

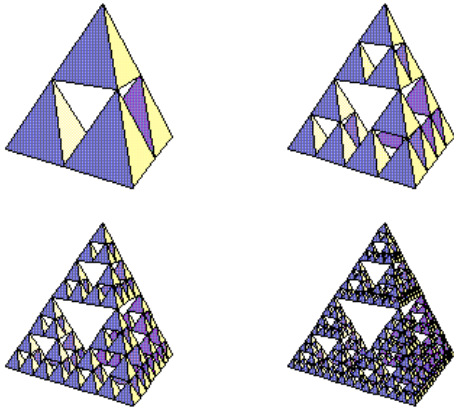
```
unsigned int phi (unsigned int n)
{
    int i, _____ (1) _____;
    for (i=2; i<n; i++)
        if ( _____ (2) _____ )
            _____ (3) _____;
    return _____ (4) _____;
}
```

ג. הסבירו מה מאפיין את אותם המספרים הטבעיים n אשר עבור מתקיים: $\varphi(n) = n-1$.

שאלה 14 (רשות)

בשיעור הכרנו צורה הנקראת משולש סרפינסקי. הצורה הבאה, הנקראת 'הפירמידה של סרפינסקי' (Sierpiński's Tetrahedron), היא הגרסה התלת-מימדית של משולש סרפינסקי:

מומלץ מאוד לבקר באתר <http://members.ozemail.com.au/~llan/Fractal.html>,



שם ניתן לשחק ביישומון (Applet) שמאפשר לכם להציג ולסובב בתלת-מימד את הצורה.

בהנחה שהצורה המקורית הייתה פירמידה משולשת (ארבעון) שאורך כל אחת מצלעותיה הוא 1. חשבו:

- א. כמה פירמידות שחורות יוצרו אחרי n איטרציות?
- ב. מה אורך צלעה של כל פירמידה שחורה שנוצרה אחרי n איטרציות?
- ג. מהו הנפח השחור הכולל שיוותר לאחר n איטרציות?

