



## מבוא למדעי מחשב מ' / ח' (234114 / 234117)

סמסטר אביב תשס"ח

### מבחן מסכם מועד א', 21 אוגוסט 2008

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
שם פרטי	שם משפחה	מספר סטודנט							

משך המבחן: 2.5 שעות.  
חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

#### הנחיות והוראות:

- מלאו את הפרטים בראש דף זה.
- בדקו שיש 22 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- כתבו את התשובות על טופס המבחן בלבד, במקומות המיועדים לכך. שימו לב שהמקום המיועד לתשובה אינו מעיד בהכרח על אורך התשובה הנכונה.
- העמודים הזוגיים בבחינה ריקים. ניתן להשתמש בהם כדפי טיוטה וכן לכתוב תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תיבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק.
- אין לכתוב הערות והסברים לתשובות אם לא נתבקשתם מפורשות לכך.
- בכל השאלות, הינכם רשאים להגדיר (ולמש) פונקציות עזר כרצונכם.
- אין להשתמש בפונקציות ספרייה, אלא אם צוין אחרת במפורש בשאלה.
- מותר להשתמש בפונקציות שמומשו בכיתה בתנאי שתצוטט החתימה המדויקת של כל פונקציה כזאת, מבחינת מספר הפרמטרים וטיפוסיהם.
- בכל שאלה ניתן להשתמש בפונקציות המוגדרות בסעיפים קודמים של אותה שאלה גם אם לא פתרם סעיפים אלו.

צוות הקורס 234114/7
<b>מרצים:</b> דר' אלי בן-ששון (מרצה אחראי), דר' רמי כהן.
<b>מתרגלים:</b> סשה סקולוזוב (מתרגלת אחראית), עלי אבוליל, אורן אשכנזי, איליה וולקוביץ', אסף פנסו, דמיטרי פציוני אריאל רביב, אייל רגב, אייל רוזנברג

שאלה	ערך	הישג	בודק
1	25		
2	25		
3	25		
4	25		
סה"כ	100		

**בהצלחה!**



- 2 -



**שאלה 1 (25 נקודות)**

סעיף א

בסעיף הזה לכל קטע קוד עליכם לציין מהי סיבוכיות הזמן וסיבוכיות המקום כפונקציה של  $n$ .

```
int fool(int n)
{
    int i, x=0, z=0;
    if(n<=1) return;
    for(i=0; i<n; i++)
        x+=n;
    for(i=0; i<x; i++)
        z+=x;

    return fool(n/2)+fool(n/2)+z;
}
```

סיבוכיות זמן:  $\Theta(n^2)$       סיבוכיות מקום נוסף:  $\Theta(\log(n))$

```
int foo2(int n)
{
    int x=0;
    while(n>0){
        int i;
        for(i=0; i<n; i++){
            x+=i;
        }
        n/=2;
    }
    return x;
}
```

סיבוכיות זמן:  $\Theta(n)$       סיבוכיות מקום נוסף:  $\Theta(1)$



- 4 -



## סעיף ב

נתונה הפונקציה הבאה:

```
int enigma (int n) {
    static int p = 0, r = 0, d = 1;

    if (p!=n) {
        p = r = n;
        d = 2;
    }
    while (d <= r) {
        if (r%d == 0) {
            r /= d;
            return d;
        }
        d++;
    }
    return 0;
}
```

1. מה ידפיס קטע הקוד הבא:

```
printf("%d ", enigma(14));
printf("%d ", enigma(6));
printf("%d ", enigma(6));
printf("%d ", enigma(6));
```

קטע הקוד הנתון ידפיס: 2 2 3 0

2. הסבירו בכמה מילים (לא יותר משתי שורות!) מה עושה קטע הקוד הבא:

```
int p;
do {
    p = enigma(n);
    printf("%d ", p);
} while (p != 0);
```

הערה: חלק 2 של סעיף ב' איננו המשך של חלק 1 של הסעיף הזה.

קטע הקוד מדפיס את כל הגורמים הראשוניים של המספר n, מן הקטן לגדול  
(עם חזרות).



- 6 -



## שאלה 2 (25 נקודות)

מערך  $a$  מגודל  $n$  נקרא **מתנדנד** (swinging) אם כל איבריו קטנים וגדלים לסירוגין.

כלומר מתקיים:  $a[i] < a[i+1]$  עבור  $i$  אי-זוגי

וגם  $a[i] > a[i+1]$  עבור  $i$  זוגי.

דוגמא למערך מתנדנד ( $n=11$ ):

1	-2	5	0	10	-1	8	7	8	4	5
---	----	---	---	----	----	---	---	---	---	---

דוגמא למערך **לא** מתנדנד ( $n=5$ ):

9	4	5	6	10
---	---	---	---	----

לשם פשטות נניח כי איברי המערך שונים זה מזה.

### סעיף א

ממשו פונקציה **swing** המקבלת מערך  $a$  ואת אורכו  $n$  ומחזירה 1 אם הוא מתנדנד ו-0 אחרת.

על הפונקציה לעבוד בסיבוכיות זמן  $O(n)$ , וסיבוכיות מקום נוסף  $O(1)$ . פתרון שלא עומד בדרישות אלו יזכה בניקוד חלקי בלבד.

```
int swing (int a[], unsigned int n) {  
    unsigned int i;  
  
    for (i = 0; i < n-1; i++) {  
        if ( ( (i % 2 == 1) && (a[i] >= a[i+1]) ) ||  
            ( (i % 2 == 0) && (a[i] <= a[i+1]) ) ) {  
            return 0;  
        }  
    }  
    return 1;  
}
```



- 8 -





### סעיף ב

ממשו פונקציה `swing_merge` המקבלת 2 מערכים **מתנדנדים** `a` ו-`b` באורך `n` כל אחד, אשר כל איבריהם שונים זה מזה, ויוצרת מערך **מתנדנד** `c` המכיל בדיוק את כל איברי `a` ו-`b`. הניחו כי אורך המערך `c` הינו  $2n$ .

על הפונקציה לעבוד בסיבוכיות זמן  $O(n)$ , וסיבוכיות מקום נוסף  $O(1)$ . פתרון שלא עומד בדרישות אלו יזכה בניקוד חלקי בלבד.

```
void swing_merge (int a[],int b[],int c[], unsigned int n) {  
    unsigned int i  
  
    for (i=0; i<n; i++) {  
        /* swing_merge_four will add four elements to  
        c, so that the last of them will be lower  
        than the highest element in the next two  
        pairs from a and b (think about it!) */  
  
        swing_merge_four(a+2*i, b+2*i, c+4*i);  
    }  
}
```



```
/* in the functions below we use the 'swap' function shown in
class: */

void swap(int *p, int* q);

/* the following is an auxiliary function for swing_merge; it
merges two swinging arrays a,b of length 2 each into an array c
of length 4, so that in the resulting sequence, the first
element is the highest of all four, and the last element is the
lowest */
int swing_merge_four (int a[], int b[], int c[]) {

    c[0] = a[0]; c[1] = a[1];
    c[2] = b[0]; c[3] = b[1];

    /* ensure the highest elements is in c[0] (the
higher elements in the two arrays are currently
in c[0] and c[2]) */
    if (c[0] < c[2])
        swap(&(c[0]), &(c[2]));

    /* ensure the lowest elements is in c[3] (the
lower elements in the two arrays are currently
in c[1] and c[3]) */
    if (c[1] < c[3])
        swap(&(c[1]), &(c[3]));
    if (c[1] > c[2])
        swap(&(c[1]), &(c[2]));
}
```



### שאלה 3 (25 נקודות)

מערך מספרים  $a$  משרה **מערך הפרשים**, שייקרא להלן  $\text{dif\_a}$ , באופן הבא:  
 $\text{dif\_a}[0] = a[0]$  כלומר  $\text{dif\_a}$  הראשון במערך ההפרשים שווה לאיבר הראשון במערך המספרים, כלומר  
עבור שאר האיברים במערך, כלומר, עבור  $i > 0$ , ערכו של כל האיבר ה- $i$  במערך ההפרשים שווה להפרש בין  
האיבר ה- $i$  לאיבר ה- $(i-1)$  במערך המקורי, כלומר  $\text{dif\_a}[i] = a[i] - a[i-1]$ .

משרה

7	5	9	17	11	15
---	---	---	----	----	----

לדוגמא, מערך המספרים הבא

7	-2	4	8	-6	4
---	----	---	---	----	---

את מערך ההפרשים הבא.

שימו לב: באמצעות מערך הפרשים ניתן לייצג מספרים שאינם בתחום הייצוג, ולכן בשאלה זו אסור להמיר מערך הפרשים למערך מספרים.

#### סעיף א

כתבו פונקציה **swap** המקבלת מערך הפרשים  $\text{dif\_a}$ , ואת אורכו  $n$ , ומחליפה את האיבר במקום ה- $i$  עם האיבר במקום ה- $i+1$ .

למשל, עבור הדוגמא לעיל. אם נקרא לפונקציה **swap**( $\text{dif\_a}$ , 6, 2) נקבל את מערך  
ההפרשים הבא:

7	-2	12	-8	2	4
---	----	----	----	---	---

הנחה: אחרי ההחלפה המספרים הינם בתחום הייצוג של מערך ההפרשים  $\text{dif\_a}$ .

על הפונקציה לעבוד בסיבוכיות זמן  $O(1)$ , וסיבוכיות מקום נוסף  $O(1)$ . פתרון שלא עומד בדרישות אלו יזכה בניקוד חלקי בלבד.

```
/* note: this function assumes i < n-1 */  
void swap (int dif_a[], unsigned int n, int i) {  
    int first_val = dif_a[i];  
    int next_val = dif_a[i] + dif_a[i+1];  
    dif_a[i] = next_val;  
    dif_a[i+1] = first_val - next_val;  
    if (i < n-2) {  
        dif_a[i+2] += next_val - first_val;  
    }  
}
```



- 12 -



### סעיף ב

כתבו פונקציה `sortDif` המקבלת מערך הפרשים ואת אורכו וממיינת אותו.  
למשל, עבור הדוגמא לעיל. אם נקרא לפונקציה `sortDif(dif_a, 6)` נקבל את מערך

5	2	2	2	4	2
---	---	---	---	---	---

ההפרשים הבא:

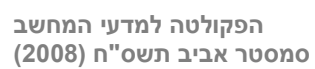
על הפונקציה לעבוד בסיבוכיות זמן  $O(n^2)$ , וסיבוכיות מקום נוסף  $O(1)$ . פתרון שלא עומד בדרישות אלו יזכה בניקוד חלקי בלבד.

```
/* We would have liked to use one of the sort functions
   we saw in class, directly, but we can't do so since
   we're not working on the actual array; note, however,
   that with clever implementation, one can rewrite
   our standard sort functions so that they can work
   for any representation whatsoever, and you would only
   need to provide a way to get element values, replace
   elements and compare elements.
   The code below is a bubble sort.
   The swap function used is the one from section Aleph.
*/
int bubbleDif (
    int dif_a[], unsigned int n) {
    unsigned int i;

    for (i=0; i<n-1; i++)
        if (dif_a[i+1] < 0)
            swap(dif_a,n,i);
}

int sortDif (
    int dif_a[], unsigned int n) {
    unsigned int i;

    for (i=0; i<n; i++)
        bubbleDif(dif_a,n-i);
}
```



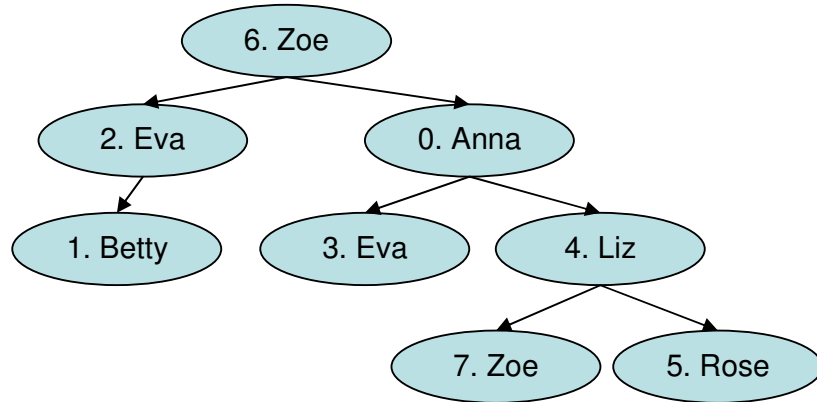
## הטכניון, מכון טכנולוגי לישראל מבוא למדעי המחשב מ"ח'

This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.



#### שאלה 4 (25 נקודות)

עץ משפחה מתאר את תולדות נשות המשפחה. לצורך הפשטות בשאלה זו איננו מתייחסים לנישואין. בדוגמא המשורטטת להלן האם-המייסדת היא Zoe אשר לה שתי בנות – Eva ו-Anna, שלוש נכדות – Liz, Betty, Eva ו-שתי נינות – Zoe ו-Rose. ל-Liz שתי בנות – Zoe ו-Rose, וכן הלאה.



הנחות: לכל אישה לכל היותר שתי בנות.  
הנחיה: בשאלה זו (בלבד!) מותר להשתמש בפונקציות ספרייה.

עץ משפחה שבה  $n$  נשים נשמר באמצעות שני מערכים:  
 $\text{char}^* \text{names}[]$  – מערך המכיל את שמות בנות המשפחה בסדר לקסיקוגרפי עולה. בשרטוט לעיל המספר מצייין את המיקום במערך השמות.  
 $\text{int tree}[n][2]$  – מערך דו-מימדי המציין לכל אישה את המיקום של שמות בנותיה במערך  $\text{names}$ . ערך (-1) מצייין אי-קיום צאצאית.

בדוגמא המתוארת בשרטוט לעיל  $n=8$  והמערכים המתאימים הם:

$\text{names} = \{ \text{"Anna"}, \text{"Betty"}, \text{"Eva"}, \text{"Eva"}, \text{"Liz"}, \text{"Rose"}, \text{"Zoe"}, \text{"Zoe"} \}$

$\text{tree} = \{ \{3, 4\}, \{-1, -1\}, \{1, -1\}, \{-1, -1\}, \{5, 7\}, \{-1, -1\}, \{0, 2\}, \{-1, -1\} \}$



- 16 -





### סעיף א

כתבו פונקציה `findOrigin` אשר מחזירה את המיקום של האם-המייסדת במערך `names`. להזכירכם, האם-המייסדת היא האישה שכל בנות המשפחה הן צאצאותיה. למשל, קריאה לפונקציה על הקלט שלעיל `findOrigin(tree, 8)`, תחזיר את הערך 6.

הנחיה: אין לשנות ערכים ב-`tree`.

```
/* we provide below a typical solution with additional space
   complexity O(n) and time complexity O(n). There also exists a
   solution with O(n) time and O(1) additional space.
*/
#define NO_ONE -1

int findOrigin (int tree[][2], unsigned int n) {
    char *has_mother;
    unsigned int i;
    has_mother = malloc(n);
    if (has_mother == NULL)
        exit(1);
    for (i=0; i<n; i++) {
        has_mother[i] = 0;
    }

    /* go over all women in the family, marking their
       children as having mothers. */
    for (i=0; i<n; i++) {
        if (tree[i][0] != NO_ONE)
            has_mother[tree[i][0]] = 1;
        if (tree[i][1] != NO_ONE)
            has_mother[tree[i][1]] = 1;
    }
    for (i=0; i<n; i++)
        if (has_mother[i] == 0) {
            free(has_mother);
            return i;
        }
}
```



```
/* we can't get here - there's always one woman with  
   no mother which we catch in the last for loop */  
free(has_mother);  
return -1;  
}
```



### סעיף ב

כתבו פונקציה `printFamily`, אשר מקבלת את המיקום של האם-המייסדת ומדפיסה את תולדות המשפחה בפורמט שלהלן. הפתרון חייב להשתמש ב**רקורסיה**.  
לכל אישה יש לציין מספר הדורות בין האם-המייסדת לבינה ואת מספר **הצאצאים** שיש לה.  
בדוגמא לעיל, הקריאה `printFamily(names, tree, 8, 6)` תדפיס את הפלט הבא:

```
Zoe belongs to generation 0. number of offspring: 7.  
Eva belongs to generation 1. number of offspring: 1.  
Betty belongs to generation 2. number of offspring: 0.  
Anna belongs to generation 1. number of offspring: 4.  
...  
Rose belongs to generation 3. number of offspring: 0.
```

הנחה: הקלט לפונקציה תקין, כלומר המערכים מייצגים עץ משפחה כמוגדר לעיל והפרמטר האחרון בקריאה לפונקציה מייצג את מיקום האם-המייסדת במערך השמות.

הנחיה: על הפתרון להיות יעיל ככל האפשר מבחינת זמן ריצה וסימפטומי. דרך ההדפסה אינו חשוב.

```
void printFamily(char* names[], int tree[][2], unsigned int n,  
                unsigned int mother) {  
    printFamily_auxiliary(  
        names, tree, n, mother, 0);  
}  
  
unsigned int printFamily_auxiliary(char* names[], int tree[][2],  
    unsigned int n, unsigned int mother, unsigned int generation) {  
    unsigned int num_offspring = 0;  
    if (mother == NOBODY)  
        return 0;  
    num_offspring +=  
        printFamily_auxiliary(  
            names, tree, n, tree[mother][0],  
            generation+1) +  
        printFamily_auxiliary(  
            names, tree, n, tree[mother][1],  
            generation+1);  
    printf("%s belongs to generation %u. Number of offspring: %u\n",  
        names[mother], generation, num_offspring);  
    return num_offspring+1;  
}
```



- 20 -



### סעיף ג

מה סיבוכיות הזמן וסיבוכיות המקום הנוסף של הפונקציה מסעיף ב'. הסבירו בקצרה.

סיבוכיות זמן:  $\Theta(\quad n \quad)$       סיבוכיות מקום נוסף:  $\Theta(\quad n \quad)$

ח הוא מס' הנשים במשפחה, וכן מספר הצמתים בעץ המשפחה.

יהי  $d$  הדור הצעיר ביותר במשפחה, הווה אומר העומק המירבי של העץ.

סיבוכיות הזמן הינה  $\Theta(n)$  שכן גוף הפונקציה מצריך זמן קבוע, וישנה קריאה אחת לכל צומת בעץ.

סיבוכיות המקום הנוסף הינה  $\Theta(d)$ , כעומק העץ, שכן גוף הפונקציה מצריך זיכרון קבוע בלי תלות

בעומק הקריאה, ועומק עץ הקריאות הוא כעומק עץ המשפחה. במקרה הגרוע, העץ יכול להיות

שרוך ארוך, ואז  $d=n$ , ובמקרה כזה הסיבוכיות הינה  $\Theta(n)$ .



- 22 -