



## מבוא למדעי המחשב מ"ח' (234117 \ 234114)

### סמסטר אביב תשע"ז

### מבחן מסכם מועד א', 12 ליוני 2017

2	3	4	1	1	
---	---	---	---	---	--

רשום/ה לקורס:

--	--	--	--	--	--	--	--	--	--	--	--

מספר סטודנט:

משך המבחן: 3 שעות.

חומר עזר: אין להשתמש בכל חומר עזר.

#### הנחיות כלליות:

- מלאו את הפרטים בראש דף זה ובדף השער המצורף, בעט בלבד.
- 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 לכתיבת תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק, פרט לדף השער אותו יש למלא בעט.
- בכל השאלות, הינכם רשאים להגדיר ולממש פונקציות עזר כרצונכם. לנוחיותכם, אין חשיבות לסדר מימוש הפונקציות בשאלה, ובפרט ניתן לממש פונקציה לאחר השימוש בה.
- אלא אם כן נאמר אחרת בשאלות, **אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה**, למעט פונקציות קלט/פלט והקצאת זיכרון (`malloc`, `free`). ניתן להשתמש בטיפוס `bool` המוגדר ב-`stdbool.h`.
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- כשאתם נדרשים לכתוב קוד באילוצי סיבוכיות זמן/מקום נתונים, אם לא תעמדו באילוצים אלה תוכלו לקבל בחזרה מקצת הנקודות אם תחשבו נכון ותציינו את הסיבוכיות שהצלחתם להשיג.
- נוהל "לא יודע": אם תכתבו בצורה ברורה "לא יודע/ת" על שאלה (או סעיף) שבה אתם נדרשים לקודד, תקבלו 20% מהניקוד. דבר זה מומלץ אם אתם יודעים שאתם לא יודעים את התשובה.
- נוסחאות שימושיות:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \Theta(\log n) \quad 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots = \Theta(1)$$

$$1 + 2 + \dots + n = \Theta(n^2) \quad 1 + 4 + 9 + \dots + n^2 = \Theta(n^3) \quad 1 + 8 + 27 + \dots + n^3 = \Theta(n^4)$$

צוות הקורס 234114/7

**מרצים:** פרופ' מירלה בן-חן (מרצה אחראית), דר' יחיאל קמחי, גב' יעל ארז **מתרגלים:** גב' דניאל עוז, גב' צופית פידלמן, מר תומר לנגה, מר יובל בנאי, דר' יוסי ויינשטיין, מר מוחמד טיבי, מר דמיטרי רבינוביץ', מר יאיר ריעאני, מר איתי הנדלר

**בהצלחה!**

[illegible]



## שאלה 1 (25 נקודות):

א. (8 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה  $f1$  המוגדרת בקטע הקוד הבא, כפונקציה של  $n$ . אין צורך לפרט שיקולים. חובה לפשט את הביטוי ככל שניתן.

```
int f1(int n) {
    int temp = n, m = n;
    while(temp)
    {
        n += m;
        temp /= 2;
    }

    for (int i=0; i < n; i++)
        for (int j=0; j < 8191; j++)
            printf("0");

    return n;
}
```

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

ב. (9 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה  $f2$  המוגדרת בקטע הקוד הבא, כפונקציה של  $n$ . אין צורך לפרט שיקולים. חובה לפשט את הביטוי ככל שניתן. הניחו שסיבוכיות הזמן של  $\text{malloc}(n)$  היא  $\Theta(1)$ , וסיבוכיות המקום של  $\text{malloc}(n)$  היא  $\Theta(n)$ .

```
int f2(int n) {
    for (int tmp=n; tmp>0; tmp/=2)
    {
        int* p = malloc(tmp);
        free(p);
    }

    for (int i=1; i*i<n; i++)
        for (int j=0; j*i<n; j++)
            printf("0");

    return n;
}
```

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

[illegible]



ג. (8 נקודות): חשבו את סיבוכיות הזמן והמקום של הפונקציה f3 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט שיקולים. חובה לפשט את הביטוי ככל שניתן.

```
#define PARTS 4

void f3(int n) {
    if (n < 4) return;

    for (int i=0; i*i<n; i++)
        printf("%d", i);

    for (int i=0; i<PARTS; i++)
        f3(n/PARTS);
}
```

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

[illegible]



## שאלה 2 (25 נק')

מטריצה ממוינת שורות היא מטריצה בה כל שורה ממוינת מהקטן אל הגדול.

למשל המטריצה הבאה היא ממוינת שורות:

1	5	7	7	9
-4	-3	0	100	101
10	100	200	201	305
0	1	2	3	5

ממשו פונקציה שחתימתה:

```
void sort(int mat[M][N], int sorted[SIZE])
```

אשר מקבלת מטריצה ממוינת שורות וממיינת את הערכים ב – mat במערך **חד-מימדי** באורך  $SIZE=N*M$  בו יהיו כל איברי המטריצה ממוינים מהקטן אל הגדול. בסיום ריצת הפונקציה על המטריצה בדוגמא למעלה, תוכן מערך sorted להיות (משמאל לימין):

-4	-3	0	0	1	1	2	3	5	5	7	7	9	10	100	100	101	200	201	305
----	----	---	---	---	---	---	---	---	---	---	---	---	----	-----	-----	-----	-----	-----	-----

ניתן להניח ש M הוא חזקה שלמה של 2, וש – M, N, SIZE מוגדרים ב - define.

תוכן מערך sorted לא ידוע בתחילת ריצת הפונקציה.

**דרישות:**

סיבוכיות זמן  $O(NM \log M)$

סיבוכיות מקום נוסף  $O(MN)$

כאשר M מספר השורות במטריצה ו N מספר העמודות (בדוגמא  $M=4, N=5$ ).

אם לפי חישוביכם לא עמדתם בדרישות הסיבוכיות אנא ציינו כאן את הסיבוכיות שהגעתם אליה:

זמן \_\_\_\_\_ מקום נוסף \_\_\_\_\_

<code>void sort(int mat[M][N], int sorted[SIZE])</code>
<code>{</code>

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



### שאלה 4 (25 נקודות) :

בתחרות עם  $n$  משתתפים  $1 - m$  שופטים, בשלב הראשון נבחרים המשתתפים אשר ימשיכו לשלב הבא. בחירת המשתתפים נעשית באופן הבא: כל שופט מבצע 3 הצבעות, כל הצבעה קובעת אם משתתף מסוים ימשיך לשלב הבא או לא. לדוגמא, שופט יחיד יכול לבצע את ההצבעות הבאות:

1. משתתף 4 ימשיך לשלב הבא
2. משתתף 2 לא ימשיך לשלב הבא
3. משתתף 19 לא ימשיך לשלב הבא

יש לבחור את המשתתפים שיעלו לשלב הבא בתחרות באופן כזה שהצבעה אחת לפחות של כל אחד מהשופטים תתקיים.

ממשו את הפונקציה:

```
int select_players(int judges[][3], int nj, int np,
                  bool chosen[])
```

מערך **judges** מכיל את הצבעות השופטים, בכל שורה יש 3 מספרי משתתפים (מספר המשתתף המינימלי הוא 1) שיכולים להיות חיוביים, כדי לסמן שההצבעה היא שהמשתתף ימשיך לשלב הבא, או שליליים, כדי לסמן שההצבעה היא שהמשתתף לא ימשיך לשלב הבא. לדוגמא, אם יש 4 שופטים ו- 5 משתתפים:

1	2	3
2	3	4
-1	-2	3
1	2	3

judges

במערך משמאל השופט הראשון והרביעי רוצים שמשתתפים 1,2,3 ימשיכו לשלב הבא, השופט השני רוצה שמשתתפים 2,3,4 ימשיכו לשלב הבא, והשופט השלישי רוצה שמשתתפים 1,2 לא ימשיכו לשלב הבא ושמשתתף 3 כן ימשיך. אף שופט לא הצביע בעד או נגד משתתף 5.

הפרמטר **nj** הוא מספר השופטים (כלומר מספר השורות במערך **judges**), והפרמטר **np** הוא מספר המשתתפים. מערך **chosen** הוא מערך בגודל **np**, ניתן להניח שכל התאים בו הם **false** כשהפונקציה נקראת לראשונה. בסיום ריצת הפונקציה, מערך **chosen** צריך להכיל **true** באינדקס  $i$  במידה ומשתתף  $i+1$  ימשיך לשלב הבא, ו- **false** אחרת (שימו לב: האינדקסים במערך מתחילים מ- 0 ומספר המשתתף המינימלי הוא 1, לכן אינדקס  $i$  במערך מתאים למשתתף מספר  $i+1$ ).

**ערך החזרה:** מספר המשתתפים שימשיכו לשלב הבא, או -1 במידה ולא ניתן לרצות את כל השופטים.

#### דוגמא לפתרון שגוי:

הפתרון שגוי כיוון שאף לא בחירה אחת של השופט השלישי מתקיימת.

true	true	false	false	false
------	------	-------	-------	-------

chosen בסיום ריצת הפונקציה

ערך החזרה: 2

#### דוגמאות לפתרונות חוקיים:

true	true	true	true	false
------	------	------	------	-------

chosen בסיום ריצת הפונקציה

ערך החזרה: 4

true	false	true	false	true
------	-------	------	-------	------

chosen בסיום ריצת הפונקציה

ערך החזרה: 3

true	true	true	true	true
------	------	------	------	------

chosen בסיום ריצת הפונקציה

ערך החזרה: 5

true	false	true	true	false
------	-------	------	------	-------

chosen בסיום ריצת הפונקציה

ערך החזרה: 3





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]