



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

### סמסטר חורף תשע"ה

### מבחן מסכם מועד ב', 16 מרץ 2015

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

רשום/ה לקורס:

#### 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 16 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- וודאו שאתם נבחים בקורס המתאים.
- 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 ומחקק, פרט לדף השער אותו יש למלא בעט.
- חובה לקרוא הוראות לכתיבת קוד המופיעות בעמוד הבא לפני פתרון המבחן.
- 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:** פרו"פ/ח תומר שלומי (מרצה אחראי), יחיאל קימחי, איתן יעקובי, אנסטסיה דוברובינה.



## הנחיות לכתיבת קוד במבחן

- בכל השאלות, הנכם רשאים להגדיר ולממש פונקציות עזר כרצונכם. לנוחיותכם, אין חשיבות לסדר מימוש הפונקציות בשאלה, ובפרט ניתן לממש פונקציה לאחר השימוש בה. מותר להשתמש בפונקציה שנכתבה בסעיף אחר, בתנאי שתציינו באופן ברור איפה הפונקציה ממומשת.
- **חובה** להקפיד על תכנות מבני (כלומר, חלוקה נכונה לפונקציות). אם כתבתם פונקציה שאורכה יותר מ 22 שורות, זוהי אינדיקציה ברורה לכך שיש לפרק את הפתרון לפונקציות עזר. אורך הפונקציה נמדד בהתאם להנחיות שניתנו בשיעורי בית.
- 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.
- ניתן להשתמש בהקצאות זיכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זיכרון.

**בהצלחה!**



## שאלה 1 (25 נקודות):

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

```
void f(int n){
    if(n>0){
        for(int k=0; k<n; k++)
            for(int j=0; j<k; j++)
                printf("1");
        f(n/2);
    }
}
```

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

ב. (9 נקודות):

```
void f(int n){
    int m = 1, k = 0;
    for (int i = 0 ; i < n ; i++){
        m *= n;
        for (int j = m ; j ; j /= 2)
            k++;
    }
}
```

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

ג. (8 נקודות): חשבו את סיבוכיות הזמן והמקום של הפונקציה  $f()$  (ולא של הפונקציה  $g$ )

```
int g(int n){
    int ctr=0;
    for (int i=1; i*i<n; ++i) {
        ctr++;
    }
    return ctr;
}
void f(int n){
    g(n*n*n);
}
```

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





[illegible]



## סעיף ב (15 נקודות)

שאר העמים בארץ התיכונה רצו לכתוב פונקציה גם בשביל השפות שלהם.

ממשו את הפונקציה הבאה, שחתימתה

```
bool middle_earth_language(char* nation, char* word);
```

הפונקציה מקבלת מחרוזת `nation` ומחרוזת `word`, ומחזירה אמת אם כל התווים במחרוזת `nation` מוכלים במחרוזת `word` (לא בהכרח עפ"י הסדר בו הם מופיעים במחרוזת `nation` ולא בהכרח פעם אחת בלבד). בסעיף זה ניתן להשתמש בלולאות. הניחו כי המחרוזות `nation` ו-`word` מכילה אותיות אנגליות קטנות.

דוגמה: על המחרוזת `feline` והעם `elf` הפונקציה תחזיר אמת, אך עבור העם `dwarf` יוחזר שקר.

**דרישות:** סיבוכיות זמן  $O(n+m)$  כאשר  $n$  זהו אורך המחרוזת `word`,  $m$  הוא אורך המחרוזת `nation`.

אם לפי חישוביכם לא עמדתם בדרישות הסיבוכיות, אנא ציינו כאן את הסיבוכיות שהגעתם אליה: \_\_\_\_\_ זמן .

```
bool middle_earth_language(char* nation, char* word)
```

```
{
```











## שאלה 4 (25 נקודות) :

בהינתן מערך חד-מימדי המכיל איברים שלמים נרצה לחלק אותו לשתי קבוצות כך שהממוצעים של איברי שתי הקבוצות יהיו שווים.

עליכם לכתוב פונקציה

```
int sameAvgGroups(int a[], int n, int res[])
```

המקבלת מערך  $a$  בגודל  $n$  ובודקת האם קיימת חלוקה שלו לשתי קבוצות (לא בהכרח רצופות ולא בהכרח שוות בגודלן) כך שהממוצעים של איברי שתי הקבוצות יהיו שווים.

אם קיימת חלוקה כזו הפונקציה תשים במערך  $res$  את האיברים של אחת מהקבוצות, ותחזיר את גודל הקבוצה. אם אין פתרון אז הפונקציה תחזיר 0 (במקרה זה אין משמעות למערך  $res$ ). המערך  $res$  מוקצה להיות באורך  $n$  מראש, אם מלאתם פחות מ  $n$  איברים אין צורך לדאוג למקום העודף.

לדוגמא, עבור המערך הבא:

5	3	12	4	12	6	1	-3
---	---	----	---	----	---	---	----

חלוקה מתאימה לשתי קבוצות תהיה:

5	4	6
---	---	---

3	12	12	1	-3
---	----	----	---	----

מכיוון שממוצע האיברים בקבוצה הראשונה הוא  $(5+4+6)/3 = 5$

ובקבוצה השנייה הוא  $(3+12+12+1-3)/5 = 5$

לכן עבור מערך כזה, אחת מאפשרויות החזרה עבור הפונקציה היא 3 והמערך  $res = \{5, 4, 6\}$

הערות:

- אסור לשנות את איברי המערך  $a$ .
- יש להשתמש בשיטת backtracking כפי שנלמדה בכיתה.
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-backtracking יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות.
- ניתן להניח שמהערך מכיל לפחות 2 איברים.



```
int sameAvgGroups(int a[], int n, int res[])
```

```
{
```







[illegible]