



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

סמסטר אביב תשע"ו

מבחן מסכם מועד א', 3 יולי 2016

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 18 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 לכתוב 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 לכתובת 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.
- שנוסחאות 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"מ מירלה בן חן (מרצה אחראית), יעל ארז, איהאב ואתד



## 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 (בלי צורך להצהיר על הפונקציה לפני). מותר להשתמש בפונקציה שנכתבה בסעיף אחר, בתנאי שתציינו באופן ברור איפה הפונקציה ממומשת.
- **חובה** להקפיד על תכנות מבני (כלומר, חלוקה נכונה לפונקציות).
- אלא אם כן נאמר אחרת בשאלות, **אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה**, למעט פונקציות קלט/פלט והקצאת זיכרון (`malloc`, `free`) והפונקציה `memcpy`. ניתן להשתמש בטיפוס `bool` המוגדר ב-`stdbool.h`.
- חתימת הפונקציה `memcpy`: `void memcpy(void *dest, void *src, unsigned size)`. שימו לב ש `size` הוא מספר **הבתיים** שצריך להעתיק.
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.

**בהצלחה!**



## שאלה 1 (25 נקודות):

א. (5 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, int arr[]){
    int i=0, j=0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}
```

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

ב. (10 נקודות): חשבו את סיבוכיות הזמן והמקום של הפונקציות  $f_1$  ו  $f_2$  כפונקציה של  $n$ .

```
int f1(int n) {
    if (n <= 1) return n;
    return 2*f1(n-1);
}

int f2(int n) {
    if (n <= 1) return n;
    return f2(n-1) + f2(n-1);
}
```

סיבוכיות זמן  $f_1$ :  $\Theta(\quad)$       סיבוכיות מקום  $f_1$ :  $\Theta(\quad)$   
 סיבוכיות זמן  $f_2$ :  $\Theta(\quad)$       סיבוכיות מקום  $f_2$ :  $\Theta(\quad)$

ג. (10 נקודות): חשבו את סיבוכיות הזמן והמקום של הפונקציה  $f()$ , כפונקציה של  $m$  ו  $n$ . ניתן להניח שסיבוכיות הזמן של  $\text{malloc}$  היא  $\Theta(1)$ .

```
void f(int n, int m){
    if (n <= 1) {
        int *arr=malloc(m*sizeof(int));
        for (int i=0; i<m; i++) arr[i] = 0;
        free(arr);
        return;
    }
    f(n-1, m+1);
    f(n%2, m+1);
}
```

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







[illegible]

[illegible]





### שאלה 3 (25 נקודות) : מחרוזות

עליכם לממש את הפונקציה:

```
int count_mixed_str(char *s1, char* s2)
```

הפונקציה מקבלת שתי מחרוזות המכילות אותיות אנגליות קטנות בלבד, וצריכה להחזיר כמה פעמים המחרוזת s1 מופיעה במחרוזת s2 ברצף אך ללא התחשבות בסדר האותיות של s1. אין לשנות את המחרוזות s1, s2.

לדוגמא:

- עבור  $s1 = "abc"$ ,  $s2 = "abcadb"$  הפונקציה תחזיר 2, שני המופעים של s1 ב-s2 הם: abcadb ו- abcadb.
- עבור  $s1 = "hi"$ ,  $s2 = "night"$  הפונקציה תחזיר 0, כי האותיות h ו-i לא מופיעות סמוכות.
- עבור  $s1 = "bba"$ ,  $s2 = "abba"$  הפונקציה תחזיר 2, שני המופעים של s1 ב-s2 הם: abba ו- abba. שימו לב שכמו בדוגמא הזאת, ייתכן שבמחרוזות יהיו כמה מופעים של אותה אות.

**דרישות:** סיבוכיות זמן  $O(m+n)$  כאשר s1 באורך m ו-s2 באורך n, וסיבוכיות מקום נוסף  $O(1)$ .

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

```
int count_mixed_str(char *s1, char* s2)
```

```
{
```









## שאלה 4 (25 נקודות) :

בהנתן מטריצה ריבועית בגודל קבוע  $N \times N$  אשר מכילה מספרים שלמים חיוביים בלבד, נגדיר "מסלול" כאוסף של תאים סמוכים. תאים סמוכים יכולים להיות שכנים מימין, משמאל, מלמעלה או מלמטה (לא באלכסון).

עליכם לממש את הפונקציה:

```
bool find_path_sum(int mat[N][N], int sum, int path[N][N]);
```

אשר מקבלת מטריצה ריבועית בגודל קבוע  $N \times N$  (ניתן להניח ש  $N$  מוגדר ב- #define), אשר מכילה מספרים שלמים חיוביים (גדולים ממש מ-0), סכום חיובי  $sum$ , ומטריצה ריבועית בגודל  $N \times N$ . הפונקציה צריכה להחזיר true במידה וקיים "מסלול" במטריצה  $mat$  עם הסכום  $sum$  – false אחרת. שימו לב שכל תא במסלול נספר פעם אחת. המטריצה  $path$ , בגודל  $N \times N$ , משמשת לסימון מסלול עם הסכום המבוקש: אם נמצא מסלול עם הסכום המבוקש,  $path$  תכיל את הערך 1 בתאים אשר משתתפים במסלול ואפסים בשאר התאים. אם לא נמצא כזה מסלול  $path$  צריכה להכיל אפסים בלבד בסיום ריצת הפונקציה. אם קיים יותר ממסלול אחד,  $path$  תכיל את אחד המסלולים עם הסכום המבוקש. לדוגמא, בהינתן המטריצה  $mat$  הבאה:

2	41	3	15
1	2	4	6
7	8	10	54
63	22	1	4

והסכום 4, הפונקציה תחזיר true והמטריצה  $path$  יכולה להיות אחת משתי האפשרויות הבאות:

0	0	0	0
0	0	1	0
0	0	0	0
0	0	0	0

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

עבור הסכום 9 לעומת זאת, הפונקציה תחזיר false –  $path$  תכיל אפסים בלבד.

עבור הסכום 3, הפונקציה תחזיר true, והמטריצה  $path$  יכולה להיות אחת מ-3 האפשרויות הבאות:

1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0

0	0	0	0
1	1	0	0
0	0	0	0
0	0	0	0

0	0	1	0
0	0	0	0
0	0	0	0
0	0	0	0



- יש להשתמש בשיטת backtracking כפי שנלמדה בכיתה.
- אסור לשנות את תוכן המערך mat.
- **מותר** להשתמש בלולאות.
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב – backtracking יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.
- מותר להשתמש בפונקציות עזר.

{

[illegible]

[illegible]





