מבוא למדעי המחשב מי/ח׳



מבוא למדעי המחשב מ'/ח' (234114 \ 234117

סמסטר אביב תש"פ

מבחן מסכם מועד א', 28 ביולי 2020

ספר סטודנט: רשום/ה לקורס: 1 1 4 3 2	2	3	4	1	1		רשום/ה לקורס:										מספר סטודנט:
---	---	---	---	---	---	--	---------------	--	--	--	--	--	--	--	--	--	--------------

משך המבחן: 3 שעות.

חומר עזר: אין להשתמש בכל חומר עזר.

הנחיות כלליות:

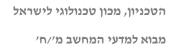
- מלאו את הפרטים בראש דף זה ובדף השער המצורף, בעט בלבד.
 - בדקו שיש 22 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- כתבו את התשובות על טופס המבחן בלבד, במקומות המיועדים לכך. שימו לב שהמקום המיועד לתשובה אינו מעיד בהכרח על אורך התשובה הנכונה.
- העמודים הזוגיים בבחינה ריקים. ניתן להשתמש בהם כדפי טיוטה וכן לכתיבת תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תיבדקנה.
 - יש לכתוב באופן ברור, נקי ומסודר, ובעט בלבד.
- בכל השאלות, הנכם 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 stdbool.h.e.
 - 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:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \Theta(\log n) \qquad \log 1 + \log 2 + \dots + \log n = \Theta(n \log n)$$
$$1^k + 2^k + 3^k + \dots + n^k = \Theta(n^{k+1}) \qquad 1 + k + k^2 + k^3 + \dots + k^n = \Theta(k^n), \ k > 1$$

צוות הקורס 234114/7

מרצים: פרופ' ניר אילון(מרצה אחראי), גב' יעל ארז מתרגלים: עמית ברכה, דן ברקוביץ', גיא ברשצקי, עמר דהרי, קטרין חדאד, דמיטרי רבינוביץ' (מתרגל אחראי)

בהצלחה!







:(שאלה 1 (25 נקודות)

סיבוכיות זמן:(_________0

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

```
int g(int n)
{
    if(n <= 1) return 1;
    return g(n / 2) + 1;
}

void f1(int n)
{
    for(int i = 1; g(i) < n; i *= 2)
    {
        printf(" * **\n");
    }
}</pre>
```

סיבוכיות מקום: (_______)❷

ב. <u>(9 נקודות)</u>: חשבו את סיבוכיות הזמן והמקום של הפונקציה f2 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט שיקוליכם. <u>חובה לפשט את הביטוי ככל שניתו.</u>

```
int g2(int n, int m)
{
    if (m >= n)
    {
        for (int i = 0; i < n; ++i)
            printf("#");
        return 1;
    }

    return 1 + g2(n / 2, 4 * m);
}

int f2(int n)
{
    return g2(n, 1);
}</pre>
```

0יבוכיות זמן:0 0 סיבוכיות מקום: 0







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

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





İ	
I	
İ	
I	
İ	
İ	
İ	
I	
İ	
İ	

הפקולטה למדעי המחשב



(נקי) אלה 2 (25 נקי)

מבוא למדעי המחשב מי/חי

ייקרא i ספרים שמכיל מספרים שלמים. המספרים שבמערך שונים זה מזה. אינדקס i ייקרא מערך ממוין a [i] = i a

ממשו את הפונקציה FindFixedPoint המקבלת מערך a של מספרים שלמים ואת אורכו מ ומחזירה את אחת מנקודת השבת של המערך (אם קיימת):

int FindFixedPoint (int a[], int n)

לדוגמה: אם נתון המערך

$\begin{vmatrix} -3 & 0 & 1 & 3 & 4 & 6 & 13 \end{vmatrix}$	12
---	----

.4 אז הפונקציה תחזיר .3 תשובה אפשרית אחרת היא

שימו לב:

- . המערך מכיל מספרים שלמים בלבד
 - המערך ממוין (בסדר עולה).
 - המספרים במערך שונים זה מזה.

:הערות

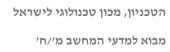
-1 אם אין נקודות שבת במערך הפונקציה תחזיר -1

:דרישות

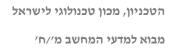
. אין הגבלה על סיבוכיות המקום. $O(\log t)$	• ממשו את הפונקציה בסיבוכיות זמן (
--	------------------------------------

אם לא עמדתם בדרישות הסיבוכיות, אנא ציינו כאן את הסיבוכיות שהגעתם אליה: זמן ______ מקום נוסף _____

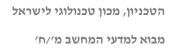
i	int	Fin	dFix	redPo	oint	(in	t a[],	int	n)			
{	•												















: (שאלה 3 (25 נקודות)

ממשו את הפונקציה:

void reverse words(char *str)

קב str המכילה מילים המופרדות ברווחים ותשנה את המחרוזת str כך str שסדר המילים יתהפך.

:דוגמא

- str[] = "geeks quiz practice code" עבור המחרוזת •
 "code practice quiz geeks" ל str אנה את הפונקציה תשנה את
 - עבור המחרוזת •

str[] = " getting good at coding needs a lot of practice"
- הפונקציה תשנה את המחרוזת ל-

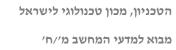
"practice of lot a needs coding at good getting "

הערות:

- ניתן להניח כי המילים מורכבות מאותיות אנגליות קטנות בלבד.
- אם מספר רווחים מפריד בין שתי מילים עוקבות במחרוזת, אז, כתוצאה מהפעלת הפונקציה,
 אותן מילים תופרדנה ע"י אותו מספר של רווחים.
- רווחים שמופיעים לפני המילה הראשונה בקלט יופיעו אחרי המילה האחרונה בפלט. בדומה,
 רווחים שמופיעים אחרי המילה האחרונה בקלט יופיעו לפני המילה הראשונה בפלט (ראו
 דוגמה).

דרישות:

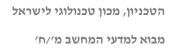
	.0(1) סיבוכיות זמן (n), כאשר n הינו אורך של המחרוזת str, סיבוכיות מקום (n).	
ליה:	לפי חישוביכם לא עמדתם בדרישות הסיבוכיות אנא ציינו כאן את הסיבוכיות שהגעתם א	אם י
	מקום נוסף	זמן
id 1	reverse_words(char *str)	













מבוא למדעי המחשב מי/חי



: (שאלה 4 (25 נקודות)

נתון מערך arr באורך n לא ממוין המכיל מספרים שלמים. ממשו פונקציה

שמקבלת את המערך arr, אורכו n ומספר שלם k ומחזירה האם קיימת חלוקה של קבוצת המספרים במערך ל- k תת-קבוצות. כך שסכום כל המספרים בכל תת-קבוצה זהה. וכך שכל איבר במערך שייך לתת-קבוצה אחת ויחידה.

לדוגמה: אם נתון מערך הבא:

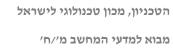
9	-2	3	17	5	3	8	4	1	1	2	
---	----	---	----	---	---	---	---	---	---	---	--

בעוד (11, 11, 15, מאחר ולא ניתן לחלק את המספרים IsKSplittable (arr, 11, 4) בעוד לארבע קבוצות העונות על כל הדרישות.

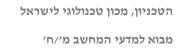
הערות:

- יש להשתמש בשיטת backtracking יש להשתמש בשיטת •
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-backtracking יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.
 - ניתן ומומלץ להשתמש בפונקציות עזר (ויש לממש את כולן).

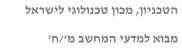
bool	l IsKSplittable(int arr[], int	nt n, int k)
{		













ĺ	
ĺ	
i	
ĺ	
ĺ	
ĺ	
ĺ	
ĺ	
ĺ	
ĺ	
ĺ	
l	
ĺ	
ĺ	
i	
ĺ	
ĺ	
ĺ	
ĺ	
i	
ĺ	

