

סמסטר חורף תשע"ח 2018

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

סמסטר חורף תשע"ח

מבחן מסכם מועד ב', 18 למרץ 2018

					 1	 	_		_	_	_	
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 20 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. ניתן בהחלט להשתמש בעיפרון ומחק, פרט לדף השער אותו יש למלא בעט.
- 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-ב המוגדר bool
 - 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:

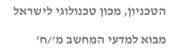
$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \Theta(\log n) \qquad 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots = \Theta(1)$$

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

צוות הקורס 234114/7

מרצים: פרופ' תומר שלומי (מרצה אחראי), פרופ' יוסף גיל, גב' יעל ארז מתרגלים: עמית אליהו, איתי הנדלר, ליאור כהן, בן לידרמן, תומר לנגה, גסוב מזאבי, נג'יב נבואני, צופית פידלמן, יורי פלדמן, עמר צברי, דמיטרי רבינוביץ' (מתרגל אחראי), יאיר ריעאני.

בהצלחה!





הפקולטה למדעי המחשב



שאלה 1 (25 נקודות):

א. $(8 \, \text{tghtim})$ חשבו את סיבוכיות הזמן והמקום של הפונקציה f1 המוגדרת בקטע הקוד הבא, כפונקציה של g. אין צורך לפרט שיקוליכם. <u>חובה לפשט את הביטוי ככל שניתן.</u>

```
void f1(int n)
{
    int i = 1, j = 0;
    while (j < n)
    {
        j += i;
        i += i;
    }
    for (; i * i < n; ++i)
        printf("*");
}</pre>
```

ב. (<u>9 נקודות)</u>: חשבו את סיבוכיות הזמן והמקום של הפונקציה £2 המוגדרת בקטע הקוד הבא,

 $\underline{\Theta}($ סיבוכיות זמן: $\underline{\Theta}($ סיבוכיות מקום: $\underline{\Theta}($

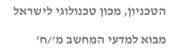
כפונקציה של n. אין צורך לפרט שיקוליכם. חובה לפשט את הביטוי ככל שניתן. הניחו שסיבוכיות $\Theta(n)$ הזמן של $\max(n)$ היא $\max(n)$, וסיבוכיות המקום של $\max(n)$.

```
int f2(int n) {
    if(n < 3)
        return 1;

    int* arr = (int*) malloc(sizeof(int) * n);
    f2(f2(n - 3));
    free(arr);

    return n;
}</pre>
```

 $\underline{\Theta}$ סיבוכיות זמן $\underline{\Theta}$ סיבוכיות מקום: $\underline{\Theta}$



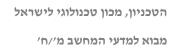




ג. (8) נקודות): חשבו את סיבוכיות הזמן והמקום של הפונקציה f המוגדרת בקטע הקוד הבא, כפונקציה של g. אין צורך לפרט שיקוליכם. <u>חובה לפשט את הביטוי ככל שניתו.</u> הניחו, כי f סיבוכיות זמן ומקום של הפונקציות g g של g בפונקציות g g של g מחזירה לוגריתם בבסיס g של g מעוגל כלפי מטה.

```
int aux(int n, int x) {
     if (x == 0)
           return n;
     int val;
     if(x % 2)
           val = aux(n * 2, x / 2);
     else
            val = aux(n / 2, x / 2);
     return val;
}
void f3(int n) {
     int m, i = log2(n);
     m = aux(4, i);
     for(i = 0; i < m; ++i)
           putchar('*');
}
```

 $\underline{\Theta}($ סיבוכיות מקום: $\underline{\Theta}($ סיבוכיות זמן $\underline{\Theta}($





1	



שאלה 2 (25 נקי)

ממשו את הפונקציה FindDuplicate המקבלת מערך n, ואת אורכו החזירה מספר כלשהו FindDuplicate המופיע במערך פעמיים או יותר. המערך לא בהכרח ממוין, ומכיל מספרים שלמים בטווח בין 1 לבין n-1.

:דוגמאות

.4 בהינתן מערך $\{1,2,3,4,5,4\}$, הפונקציה תחזיר

בהינתן מערך $\{1,1,1,2,2,2\}$, הפלטים האפשריים של הפונקציה הם 1 או 2.

דרישות:

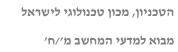
- . $\underline{\mathbf{0}}(1)$ על הפונקציה לעמוד בסיבוכיות זמן של , $\underline{\mathbf{0}}(n)$ על הפונקציה לעמוד
- פתרונות בסיבוכיות זמן/מקום גרועה מהנדרש יזכו לניקוד חלקי במידה והם נכונים.

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

שימו לב: ניתן לשנות את איברי המערך.

זמן מקום נוסף

<pre>int FindDuplicate(int a[], int n)</pre>
{















: (שאלה 3 (25 נקודות)

. מחרוזת תקרא א-שופעת אם לפחות תו אחד בה חוזר על עצמו ${f k}$ פעמים או יותר

 ${\sf e}$ למשל, ${\sf deadbeef}$ היא מחרוזת 3-שופעת, אך לא מחרוזת 4-שופעת, מאחר והתו הכי נפוץ בה מופיע שלוש פעמים.

.שופעת, אך לא -k-שופעת, אך לא א-שופעת. אם היא (-1)-שופעת.

בדוגמה לעיל, המחרוזת היא 4-חסרה.

ממשו פונקציה GetKShortSubstring שמקבלת מחרוזת str שמקבלת שמקבלת התת-מחרוזת הערכה אורך התת-מחרוזת k נתון. אם תת-מחרוזת כזו לא קיימת הפונקציה תחזיר k נתון.

למשל, בדוגמה שמלווה אותנו, בעבור k=3, הפונקציה תחזיר 6, כי משל, בדוגמה שמלווה אותנו, בעבור k=3, האונה על הדרישות היא adbe וגם eadb ולכן הפונקציה תחזיר k=2 תת-מחרוזת העונה על הדרישות היא k=2 חסרה. k=2

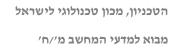
דרישות:

- על הפונקציה לעמוד בסיבוכיות זמן 0(n) , וסיבוכיות מקום 0(1), כאשר 0 הוא אורך מחרוזת הקלט.
 - אפשר להניח שכל האותיות במחרוזת הקלט הן קטנות.
 - פתרונות בסיבוכיות זמן/מקום גרועה מהנדרש יזכו לניקוד חלקי במידה והם נכונים.
 - שימו לב, כי תת-מחרוזת זהו **רצף** של תווים.

<pre>int GetKShortSubstring(char *str, int k)</pre>
{













L	



: (שאלה 4 (25 נקודות)

טקס האוסקר מתקרב, וכל מוזמנת צריכה לבחור צבע לשמלה. ממשו פונקציה המקבלת נתונים על העדפות צבע השמלה של N מוזמנות לטקס ומחזירה את מספר הצבעים המינימלי הדרוש לפי הכללים בכאים:

כל מוזמנת מציינת קבוצה של מוזמנות אחרות שצריך להיות להן את אותו צבע שמלה כמו שלה, $color_map[N][N][N]$, הדו ממדי [M] $color_map[N][N][N]$, ערך בשורה ה i במקום ה j, ערך של 1 מציין דרישה של מוזמנת i לאותו צבע שמלה כמו למוזמנת j, ערך של 1- מציין דרישה לצבע שונה, וערך 0 מציין שאין דרישה כלשהי.

ערך החזרה: מספר הצבעים המינימלי הדרוש, העונה על הדרישות (או 1- במקרה שאין כזה).

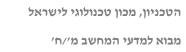
הערות:

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

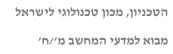
<pre>int FindMinimumNumDressColors(int color_map[N][N])</pre>		
{		













1	

