

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

סמסטר אביב תש"ע מבחן מועד ב' - 16.9.2010

שם פרטי	שם משפחה				טודנט	פר ס	מספ
	: השלים אם צריך	ז ול	ות הרלוונטיו	אפשר	את ר	'סמן	נא ל
	234117	/	234114	;ורס:	ם/ה לק	רשונ	•
	: אחר (לפרט)	/	לימודי חוץ	/ J II	ר ראש	תואו	•
				ת:	הוראו	יות ו	הנח

אין להשתמש בכל חומר עזר.

משך המבחן: 3 שעות.

- מלאו את הפרטים בראש דף זה.
- בדקו שיש 24 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- כתבו את התשובות לשאלות על טופס הבחינה בלבד. השתדלו לכתוב את תשובתכם במקום הריק מייד לאחר השאלה. במידה ותשובתכם נמצאת בעמוד אחר, אנא ציינו את מספר העמוד בסמוך לשאלה.
- ניתן להשתמש בדפים הריקים בטופס הבחינה גם כדפי טיוטה. הקפידו לסמן טיוטות באופן ברור על מנת שהן לא תיבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר, ולהקפיד על עימוד והזחה (אינדנטציה) של הקוד כפי שנלמד בכיתה.
 ניתן בהחלט להשתמש בעיפרון ומחק, פרט לדף השער אותו יש למלא בעט.
 - בכל השאלות, הינכם רשאים להגדיר (ולממש) פונקציות עזר כרצונכם, אלא אם מצויין אחרת בשאלה.
- אין להשתמש בפונקציות ספריה, או בפונקציות שמומשו בכיתה אלא אם צוין אחרת במפורש בשאלה (פרט לפונקציות קלט/פלט והקצאת זיכרון).
- בכל שאלה ניתן להשתמש בפונקציות שמוגדרות בסעיפים קודמים של אותה שאלה גם אם לא פתרתם סעיפים אלו.

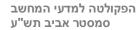
בודק	הישג	ערך	שאלה
		15	1
		30	2
		30	3
		25	4
		100	סה"כ

צוות הקורס 234114/7

מרצים: ד"ר תומר שלומי (מרצה אחראי), רובינשטיין.

מתרגלים: ישראל גוטר (מתרגל אחראי), סינטיה דיסנפלד, אופיר ובר, מירב זהבי, מרינה סקרבובסקי, רועי פורן, אייל רגב, נדב שרגאי.

בהצלחה!





l	



<u>שאלה 1 (15 נקודות)</u>

בשאלה זו 3 סעיפים בלתי תלויים זה בזה.

<u>סעיף א</u>

```
#define K 5
int f1(int n)
{
   int s=0;
   int d, m;
   while (n>1)
   {
      scanf("%d", &m);
      d = (m > 2*K) ? 2*K : (m < K ? K : m);
      n /= d;
      s += n;
   }
   return s;
}</pre>
```

(ניתן להתייחס ל-לK כקבוע) מהי סיבוכיות הזמן והמקום של f1 כפונקציה של

O (________) פיבוכיות זמן: (_______) O (סיבוכיות מקום נוסף: (

הפקולטה למדעי המחשב סמסטר אביב תש"ע





<u>סעיף ב</u>

```
int f2(int n)
{
   int j, k, cnt=0;
   do {
       ++n;
   } while (n%3);
   for (j=n; j>0; j/=3)
   {
       k=j;
       while (k>0)
       {
            cnt++;
            k-=3;
       }
   }
   return cnt;
}
```

2n מהי סיבוכיות הזמן והמקום של 12 כפונקציה של

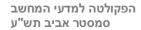
Θ ()	סיבוכיות מקום נוסף:	Θ(סיבוכיות זמן: (

<u>סעיף ג</u>

```
int f3(int n, int m)
{
   if (n<1 || n>m) return n;
   if (n%5==0) return f3(n+5, m);
   return f3(n+1, m);
}
```

m-1 n ו-m מהי סיבוכיות הזמן והמקום של

```
\Theta ( _____ ) \Theta סיבוכיות זמן: ( _____ ) \Theta
```





-	



<u>שאלה 2 (30 נקודות)</u>

מטריצה **מסודרת שורות** הינה מטריצה עם החוקיות הבאה: כל האיברים בכל שורה גדולים ממש מכל האיברים בשורה שאחריה, וכך הלאה.

: למשל, המטריצה הבאה (4x6) מסודרת שורות

a[i][j]	j					
	73	65	70	68	66	54
	38	37	35	43	40	50
li	20	16	19	14	33	31
	7	4	3	8	12	10
₩						

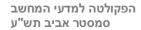
סעיף א (15 נקודות)

עליכם לממש פונקציה שכותרתה

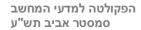
אם x ומספר (#define-מוגדר כ-N) $N^2 \times N$ ומספר x, ומחזירה 1 אם x הפונקציה מקבלת מטריצה מסודרת שורות בגודל x מצא במטריצה ו-0 אחרת. כמו כן אם x נמצא במטריצה, הפונקציה כותבת את הקואורדינאטות שלו למצביעים x ו-x למצביעים x ו-x למצביעים x ו"

O(1) וסיבוכיות מקום נוסף O(N) ארישות סיבוכיות: יש לפתור בסיבוכיות זמן

. הערה: שימו לב שניתן (ומומלץ) לממש פונקציות עזר על מנת לפשט את הקוד









·	





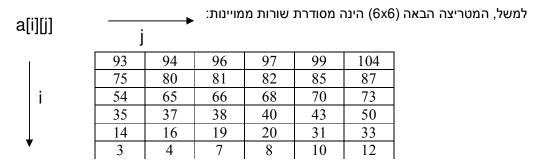
-	



סעיף ב (15 נקודות)

מטריצה מסודרת שורות ממויינות הינה מטריצה שבה:

- 1. כל שורה מכילה איברים גדולים ממש מאברי השורה הבאה.
 - 2. כל שורה ממוינת בסדר עולה, משמאל לימין.



עליכם לממש פונקציה שכותרתה

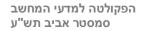
void merge2d(int a[N][N], int b[N][N], int c[2*N][N])

הפונקציה מקבלת שתי מטריצות מסודרות שורות ממויינות [N][N] ו-[N][N], וכן מטריצה ריקה [N][N] .c[2*N][N] הפונקציה ממזגת את תוכן a ו-b לתוך c כאשר c צריכה להיות מטריצה מסודרת שורות ממויינות גם כן. כלומר, בסוף התהליך c צריכה להיות מטריצה מסודרת שורות ממויינות המכילה את כל אברי a ו-b.

.O(1) וסיבוכיות מקום (O(N^2) דרישות סיבוכיות: יש לפתור בסיבוכיות זמן b -טיבוכיות מקום b -טיבוכיות ביחד בל שתי המטריצות ביחד ביחד מזה.









[
[
[
- <u></u>



<u>שאלה 3 (30 נקודות)</u>

(סעיף א (15 נקודות)

נתונים שני מערכים <u>לא ממוינים</u> של מספרים שלמים, []a באורך n1 ו-[b] באורך n2. בשני המערכים ייתכנו ערכים שמופיעים יותר מפעם אחת. עליכם לממש פונקציה רקורסיבית שכותרתה

```
int intersect(int a[], int n1, int b[], int n2)
```

הפונקציה מקבלת כפרמטרים את שני המערכים ואת אורכיהם, ומחזירה כפלט את מספר האיברים <u>השונים</u> <u>זה מזה</u> (ללא חזרות) המשותפים לשני המערכים. למשל, עבור המערכים

```
b[] = \{2,1,1,5\} \qquad a[] = \{5,3,1,1,5,8\}
```

המספרים המשותפים הינם 1 ו-5 (לא סופרים חזרות) ולכן הפונקציה מחזירה 2.

:הערות

1. לצורך השאלה נתונה פונקצית העזר

```
int exist(int x, int a[], int n)
```

המקבלת מערך a (לא בהכרח ממוין) באורך n, ומספר x, ומחזירה 1 אם x נמצא ב-a, או 0 אחרת. סיבוכיות α המקבלת מערך α , וסיבוכיות המקום $\Theta(n)$, וסיבוכיות המקום $\Theta(n)$

- 2. על הפונקציה intersect להיות רקורסיבית. <u>אין להשתמש בכל פונקצית עזר</u> פרט לפונקציה exist.
 - 3. יש להשתמש בשלד הבא עבור הפתרון:

```
int intersect(int a[], int n1, int b[], int n2) {
    ...
    if (exist(a[0],a+1,n1-1)) {
        ...
    }
    else {
        ...
    }
    ...
}
```

4. השלימו את סיבוכיות הזמן והמקום של הפונקציה שלכם כתלות ב-n1 ו-n2:

```
\Theta ( ______ ) סיבוכיות זמן: ( ______ ) \Theta סיבוכיות מקום נוסף: \Theta
```

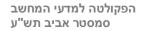




הפקולטה למדעי המחשב סמסטר אביב תש"ע



l
l





l
l



<u>סעיף ב (15 נקודות)</u>

ממשו פונקציה רקורסיבית שכותרתה

int find_one(int a[], int n1, int b[], int n2)

הפונקציה מקבלת מערכים [a] ו-[b] באורכים n1 ו-n2 בהתאמה, שאינם ממוינים. הפונקציה מחזירה אינדקס של ערך כלשהו שנמצא ב-a אך לא נמצא ב-b. במידה ואין ערך כזה, הפונקציה מחזירה 1-. ניתן להניח שערכים אינם חוזרים על עצמם באף אחד מהמערכים.

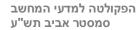
בסעיף זה חלה הדרישה הבאה:

אין לבצע בקוד השוואות מפורשות בין האברים של המערכים [] או [b] וכן אין לגשת לאף אחד מאברי המערכים ישירות באמצעות האופרטורים [] או *. עם זאת, לצורך הפתרון ניתן להשתמש בפונקציה המערכים ישירות באמצעות האופרטורים [] או *. עם זאת, לצורך הפתרון ניתן להשתמש בפונקציה intersect() מהסעיף הקודם. לדוגמה, ניתן לבדוק אם האיבר [3] נמצא במערך [b] ללא גישה מפורשת לאיבר זה על ידי הקריאה

if $(intersect(a+3,1,b,n2)==1) \{ ... \}$

.intersect() קריאות לפונקציה O(logn1) דרישות סיבוכיות: על הפתרון לבצע לכל היותר

שימו לב שהפתרון נדרש להיות רקורסיבי.





-	
	-



<u>שאלה 4 (25 נקודות)</u>

ממשו פונקציה שכותרתה

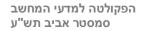
int find_path(int price[N][N], int x, int path[N])

הפונקציה מחפשת מסלול שעובר בין כל האיים, כך שמבקרים <u>בכל אי בדיוק פעם אחת</u>. כמו כן, העלות היא הכוללת של המסלול צריכה להיות לכל היותר x. אם קיים מסלול כזה, הפונקציה מחזירה 1, אחרת היא מחזירה 0. כמו כן, במידה ונמצא מסלול, על הפונקציה לכתוב אותו לתוך המערך [path[N]. על המערך [path לאי הכיל בתא ה-i את <u>מספר הצעד שבו ביקרנו באי ה-i</u>. למשל, אם המסלול שנמצא מתחיל באי 3 וממשיך לאי path[3] הו-2=[7]path ו-2=[7] הלאה לכל האיים במסלול.

<u>הערות:</u>

- 1. על הפתרון להיות רקורסיבי ולהשתמש ב backtracking.
- 2. ניתן להניח כי המערך [|path מאותחל לאפסים בקריאה הראשונה לפונקציה find_path.

—
_
_
_
—



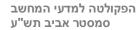






-	





-