

מבוא למדעי מחשב מ' / ח' (234114 / 234117) סמסטר אביב תשע"ג

מבחן מסכם מועד א', 11 יולי 2013

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 14 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 תבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. <u>ניתן בהחלט להשתמש בעיפרון ומחק,</u> פרט לדף השער אותו יש למלא בעט.
- בכל השאלות, הינכם 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.-a המוגדר ב-bool
 - אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- ניתן להשתמש בהקצאות זכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זכרון.
- 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.
 - 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' ניר אילון (מרצה אחראי), ד"ר תמיר לוי, חביאר טורק.

מתרגלים: נחשון כהן, נדיה לבאי, עלי חמוד, אביב סגל, ירון קסנר, ורד כהן, אנסטסיה דוברובינה (מתרגלת אחראית).

בהצלחה!





שאלה 1 (25 נקודות):

א. (12) המוגדרת בקטע הקוד המקום של הפונקציה f המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט שיקוליכם.

int	f(int n)					
{						
	if $(n == 1)$					
	return 1;					
	return $f(f(n-1)$);				
}						
	Θ (כיות מקום:	סיבו	Θ ()	סיבוכיות זמן:
		•		-		•
	ודם:	מהסעיף הק	ת הפונקציה	ה את הגדרו	נ) נשנה עת	ב. (13 נקודות
			-			
int	f(int n)					
int {	f(int n)		•			· · ·
int {	f(int n) if (n == 1)					· ·
int {						· ·
int {	if (n == 1)					· ·
int {	if (n == 1) return 1;	(n-1));				· -
int {	if (n == 1)	(n-1));				
int {	if (n == 1) return 1;	(n-1));				· -
int { }	if (n == 1) return 1;	(n-1));	הקלנו מ?			מה ערך החזרו







שאלה 2 (25 נקודות):

<u>(סעיף א (12 נקודות)</u>

ממשו את הפונקציה הבאה, שחתימתה

int limit and sort(int a[], int n, int m, int p[]);

m>1 הפונקציה מקבלת כקלט מערך a של מספרים **שלמים וחיוביים** באורך n, עוד מספר a וכן מצביע p למערך של לפחות m תאים (**המערך כבר מוקצה**). על הפונקציה למצוא את כל p המספרים ב- a שערכם לכל היותר m ולכתוב אותם בצורה ממוינת ובלי חזרות למערך a. על הפונקציה להחזיר את מספר המספרים שנכתבו ל-p. לדוגמא, אם a הוא המערך הבא:

int $a[7] = \{4, 7, 5, 4, 3, 30, 201\};$

וכן n=7, m=200 את על הפונקציה לכתוב לתוך 5 התאים הראשונים של המערך n=7, m=200 התוכן הבא (משמאל לימין):

3 4	5	7	30
-----	---	---	----

וכן להחזיר את המספר 5.

O(m) וסיבוכיות מקום נוסף (O(m+n) וסיבוכיות מקום נוסף

אנא ציינו כאן את הסיבוכיות שהגעתם	מדתם בדרישות הסיבוכיות,	אם לפי חישוביכם לא עו
	מקום נוסף	אליה: זמן

int limit_and_sort(int a[], int n, int m, int p[])
{





סעיף ב (13 נקודות)

בהינתן מערך a של מספרים חיוביים, נאמר שמספר z הוא *מיוחד ביחס ל-* a אם קיימים שני a בהינתן מערך x-y ש z מתחלק ב- x.y שנמצאים שניהם ב- a כך ש z מתחלק ב-

int
$$a[7] = \{4, 7, 5, 4, 3, 30, 201\}$$

אז המספר 24 הוא מיוחד ביחס ל- a כי הוא מתחלק ב- 3.4=12 והמספרים a במערך. לעומת זאת, 25 אינו מיוחד ביחס ל- a כי הוא אינו מתחלק במכפלה של שני מספרים שונים a ב- a.

ממשו פוקנציה המקבלת מערך a, את גודלו n, ומספר נוסף m>1. על הפונקציה להדפיס את a, את גודלו a, את גודלו a, המיוחדים ביחס ל- a, יש להדפיס את המספרים בסדר עולה ואין להדפיס אף מספר פעמיים.

O(m) וסיבוכיות מקום נוסף ($O(n+m \cdot \log^2 m)$ וסיבוכיות מקום נוסף

<u>ניתן ומומלץ</u> להשתמש בפונקציה מהסעיף הקודם, גם אם לא מימשתם אותה.

ת הסיבוכיות שהגעתם	אנא ציינו כאן א	דרישות הסיבוכיות,	לא עמדתם ב	י חישוביכם	אם לפ
		קום נוסף	מ		:אליה

<pre>void print_specials(int a[], int n, int m)</pre>
{





שאלה 3 (25 נקודות):

עליכם לממש את הפונקציה:

void swap first last(char arr[])

הפונקציה מקבלת כקלט מערך של תוים שמכיל **שלוש** מחרוזות אחת אחרי השניה. לדוגמא:

על הפונקציה להחליף בין המחרוזת הראשונה לשלישית. למשל, בדוגמא לעיל, לאחר הפעלת הפונקציה המערך יכיל את התוכן הבא:

דרישות: סיבוכיות זמן O(n) וסיבוכיות מקום נוסף O(n), כאשר n הוא מספר התוים הכולל O(n) שימו לב: תניחו שהמערך arr מכיל בדיוק שלושה תווי סיום מחרוזת ordetarr במערך ordetarr שימו לב: תניחו שהמערך ordetarr שימו לב: תניחו שהמערך ordetarr מכיל בדיוק שלושה תווי סיום מחרוזת ordetarr והם ניספרים בהגדרה של ordetarr (ordetarr).

void reverse (char v[], int k); מותר לכם להשתמש בפונקצית עזר k אווים. דוגמא: אם נגדיר במערך נתון k של k תווים במערך נתון של אווים.

char $v[4] = \{ 'a', 'b', ' \setminus 0', '5' \}$

 $v \cdot v$ תשנה את תוכן המערך רישנה את reverse (v, $v \cdot d$) אז הפעלת

פועלת בסיבוכיות (k) פועלת וועלת פייבוכיות פונקציה reverse פונקציה. { '5', '\0', 'b', 'a'}

. אין צורך לממש אותה. (1)

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

<pre>void swap_first_last(char arr[])</pre>
{





: (שאלה 4 (25 נקודות)

קליקה היא קבוצה של אנשים שבה כל זוג אנשים הם חברים. נתון מערך דו מימדי של friends [i] [j] . int friends [N] [N] . ברויות . int friends [n] . הוא 1 אם . i . חברים, . ו-0 אחרת. המערך סימטרי, כלומר מתקיים friends . int friends . int friends . interval . int

עליכם לכתוב פונקציה

int hasClique(int friends[][N], int k)

המקבלת מערך חברויות friends, ומספר ${\bf k}$. הפונקציה תחזיר 1 אם קיימת קליקה המקבלת מערך אנשים, ו-0 אם לא קיימת קליקה כזו.

לדוגמא, עבור מטריצת החברויות הבאה:

	0	1	2	3
0		1	1	0
1	1		1	0
2	1	1		1
3	0	0	1	

קבוצת החברים {0,1,2} היא קליקה בגודל 3, וקבוצת החברים {2,3} היא קליקה בגודל 2, אך לא קיימת קליקה בגודל 4 (כי לדוגמא 3 ו-0 לא חברים).

:הערות

- .#define מוגדר ע"י N ●
- יש להשתמש בשיטת backtracking כפי שנלמדה בכיתה.
- יש לוודא שלא backtracking-בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-מקובל מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.

<pre>int hasClique(int friends[][N], int k)</pre>
{





