



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

סמסטר חורף 2008-2009

מבחן מסכם מועד א', 19 פברואר 2009

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
שם פרטי	שם משפחה	מספר סטודנט							

משך המבחן: 3 שעות.
חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

הנחיות והוראות:

- מלאו את הפרטים בראש דף זה.
- בדקו שיש 26 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 לכתוב תשובותיכם. סמנו טיוטות באופן ברור על מנת שהן לא תיבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק.
- אין לכתוב הערות והסברים לתשובות אם לא נתבקשתם מפורשות לכך.
- בכל השאלות, הינכם רשאים להגדיר (ולממש) פונקציות עזר כרצונכם.
- אין להשתמש בפונקציות ספריה, או בפונקציות שמומשו בכיתה אלא אם צוין אחרת במפורש בשאלה למעט פונקציות קלט פלט והקצאת זיכרון (malloc).
- בכל שאלה ניתן להשתמש בפונקציות המוגדרות בסעיפים קודמים של אותה שאלה גם אם לא פתרתם סעיפים אלו.

צוות הקורס 234114/7
מרצים: פרופ' חבר מיקי אלעד (מרצה אחראי), דר' רמי כהן, מר תמיר לוי, מר איתי שרון.
מתרגלים: דן רביב (מתרגל אחראי) אייל רגב, אורן אשכנזי, מירלה בן-חן, אילייה וולקוביץ, רון בגלייטר, דויד וייץ, יוחאי קפלן, אופיר וובר, דודו ינאי, רועי פורן, אייל רוזנברג, רועי אדדי. בודקי שיעורים: קיילה ג'קובס, עלי אבו-ליל.

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

בהצלחה!



- 2 -



שאלה 1 (15 נקודות)

סעיף א (10 נקודות)

תארו בקצרה מה עושה הפונקציה הבאה ומה היא מחזירה.

```
int foo(int a[], int n, int x)
{
    int r;
    int d;
    if(n<=0) return -1;
    r=abs(a[0])%n; /* the function abs() returns the absolute value of
the argument in time complexity O(1) */

    if(a[r]==x) return r;
    if(a[r]>x){
        d=foo(a, r, x);
        return d;
    }
    if(a[r]<x){
        d=foo(a+r+1, n-r-1, x);
        if(d!=-1) return d+r+1;
        return -1;
    }
}
```

מהי סיבוכיות הזמן והמקום של הפונקציה? נמקו בקצרה.

סיבוכיות זמן: $\Theta(\text{_____})$ סיבוכיות מקום נוסף: $\Theta(\text{_____})$



- 4 -





- 6 -



שאלה 2 (30 נקודות)

נתונה מטריצה בגודל $N \times N$ המכילה מספרים שלמים. בהיתנן שתי תת-מטריצות ריבועיות נאמר שהן "זהות בסיבוב" אם כל איבריהן זהים בהתאמה או אם ע"י סיבוב של אחת המטריצות ב- 90° או ב- 180° או ב- 270° נקבל שתי תת מטריצות שכל איבריהן זהים בהתאמה.

לדוגמא, במטריצה הבאה שגודלה 5×5 , שתי תת-המטריצות המודגשות "זהות בסיבוב", כי אם נסובב אחת מהן ב- 90° עם כיוון השעון אז איבריהן יהיו זהים.

1	7	12	5	1
18	3	5	13	3
9	2	9	18	1
13	16	2	3	7
6	4	12	9	11

דוגמא נוספת: תת המטריצות המודגשות זהות בסיבוב של 180°

1	7	12	5	19	6	4	1
18	3	5	13	4	8	7	3
9	2	9	18	9	8	1	1
13	16	2	3	2	5	9	7
13	4	8	7	3	2	16	13
18	9	8	1	18	9	2	9
3	2	5	9	13	5	3	18
9	13	15	21	5	12	7	1



- 8 -



- 10 -



סעיף ב (10 נקודות)

כיתבו פונקציה

```
int find_identical(int a[N][N], int k)
```

המקבלת מטריצה המיוצגת ע"י מערך דו מימדי a , ומספר שלם k , ובודקת האם יש ב- a לפחות שתי תת מטריצות (שאינן חופפות באופן מלא, כי הרי ברור שכל תת מטריצה "זהה בסיסוב" לעצמה) ריבועיות שגודל כל שורה ועמודה הוא k , שהן "זהות בסיבוב". אם כן הפונקציה מחזירה 0 ואם לא היא מחזירה 1.

סיבוכיות מקום נוסף נדרשת $O(1)$, סיבוכיות זמן נדרשת $O(N^4 \cdot k^2)$

```
int find_identical(int a[N][N], int k){
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



- 12 -

int max_identical(int a[N][N]) מקבלת מטריצה המיוצגת ע"י מערך דו מימדי a, ומחזירה את גודלה של תת-המטריצה הריבועית (כלומר שגודל השורה שווה לגודל העמודה) הגדולה ביותר שעבורה יש תת-מטריצה (שונה) "זהה בסיבוב".
על הפונקציה לעבוד בסיבוכיות מקום נוסף של $O(1)$ ובסיבוכיות זמן יעילה יותר מ- $\Theta(N^7)$.

```
int max_identical(int a[N][N]){
```



- 14 -



שאלה 3 (30 נקודות)

תמורה (פרמוטציה) של סדרת מספרים היא סידור כלשהו של כל אברי הסדרה.
פרמוטציה של אינדקסים היא מערך מספרים, שאבריו הם כל האינדקסים האפשריים של המערך בסדר כלשהו
(זיכרו: האינדקסים של מערך בגודל n הם $[0, n-1]$)

למשל, המערך הבא הוא פרמוטציה של אינדקסים

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

ואילו המערכים הבאים אינם פרמוטציות של אינדקסים

3	0	3	1
---	---	---	---

4	0	5	1
---	---	---	---



- 16 -



סעיף א (10 נקודות)

כתבו פונקציה

`int indexPermutation(int a[], int n)`, המקבלת מערך של `a` ואת אורכו `n`, ובודקת האם הוא פרמוטציה של אינדקסים. אם כן, הפונקציה מחזירה 0 ואם לא, הפונקציה מחזירה 1.

סיבוכיות זמן נדרשת $O(n)$.

```
int indexPermutation (int a[], int n){
```



- 18 -



- 20 -



- 22 -



שאלה 4 (25 נקודות)

ניתן לתאר מערכת כבישים בין ערים באמצעות מטריצה a שבה ערכו של האיבר $a[i][j]$ הוא כאורך הכביש הישיר המחבר את העיר i עם העיר j . אם לא קיים כביש המחבר את i עם j יהיה ערכו של האיבר $a[i][j] = -1$.

לדוגמא, המטריצה הבאה מייצגת מערכת כבישים בין 3 ערים. במערכת זו אין כביש המחבר את העיר 0 עם העיר 1 (כי $a[0][1] = -1$) ואילו אורכו של הכביש המחבר את העיר 0 עם העיר 2 הוא 7 (כי $a[0][2] = 7$).

0	-1	7
3	0	1
5	4	0

שימו לב שהכבישים הם חד סיטריים, כלומר אורכו של הכביש מ- i ל- j אינו שווה בהכרח לאורך הכביש מ- j ל- i .

כיתבו פונקציה **רקורסיבית** `int find_path(int a[N][N], int s, int t, int k, int path[])` המקבלת מערכת כבישים המיוצגת ע"י מערך דו-מימדי a וכמו כן מקבלת את האינדקס של עיר המוצא s ועיר היעד t . הפונקציה בודקת האם קיים מסלול נסיעה מהעיר s אל העיר t שאורכו אינו גדול מ- k . אם קיים מסלול כזה הפונקציה מחזירה את מספר הערים במסלול (כולל s ו- t) ואת המסלול עצמו (במערך `path`). אם לא קיים מסלול כזה, הפונקציה מחזירה -1. הניחו כי אורכם של כל הכבישים גדול ממש מ-0, והמרחק של עיר מעצמה הינו 0 (כלומר $a[i][i] = 0$). הניחו כי N מוגדר באמצעות `#define`.

שימו לב: פתרון שאינו רקורסיבי לא יקבל ניקוד



- 24 -



- 25 -



- 26 -