



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

סמסטר אביב תשס"ו

מבחן מסכם מועד ג', 3 נובמבר 2006

<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 22 עמודים (5 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.
- אין לכתוב הערות והסברים לתשובות אם לא נתבקשתם מפורשות לכך.
- בכל השאלות, הינכם רשאים להגדיר (ולממש) פונקציות עזר כרצונכם.
- אין להשתמש בפונקציות ספרייה או בפונקציות שמומשו בכיתה אלא אם צוין אחרת בשאלה.
- פתרון שלא עומד בדרישות הסיבוכיות יקבל ניקוד חלקי בלבד.

צוות הקורסים 234114/7
מרצים: סאהר אסמיר, פרופ' חיים גוטסמן (מרצה אחראי).
מתרגלים: אייל רוזנברג, מירי בן חן, אולג רוכלנקו, רן רובינשטיין, שיאון שחורי, שי אוחיון (מתרגל אחראי).
בודקי תרגילים: מרק גינזבורג, מאשה ניקולסקי

שאלה	ערך	הישג	בודק
1	20		
2	20		
3	20		
4	20		
5	20		
סה"כ	100		

בהצלחה!



- 2 -



A large rectangular area containing horizontal lines for writing, intended for the student's answer.



- 6 -



שאלה 2 (20 נקודות)

נתונה הפונקציה הבאה:

```
void func(unsigned int n, unsigned int m) {  
    unsigned int i;  
  
    if (n>=m) {  
        while (m)  
            m/=2;  
        return;  
    }  
  
    func(2*n, m);  
    func(2*n, m);  
  
}
```

סעיף א

1. כתבו את המשוואה הרקורסיבית המתארת את סיבוכיות הזמן של הפונקציה כתלות ב- n וב- m .
ניתן להניח כי $m > n > 1$.

$T(n, m) =$ _____

2. מהו תנאי העצירה של הרקורסיה?

3. מהי סיבוכיות הזמן של הפונקציה עבור תנאי העצירה?

סעיף ב

פתרו את המשוואה הרקורסיבית שבסעיף א' עבור T עם ערכי n, m כלליים. מצאו את החסם ההדוק.
ניתן להניח כי $m > n > 1$. (כתבו את כל השלבים לפתרון, ולא רק התוצאה הסופית)

<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>



- 8 -



שאלה 3 (20 נקודות)

מערך אוורירי הינו מערך המאפשר פעולת הכנסת איבר חדש בין איברים קיימים ב- $O(1)$. העבודה עם מערך אוורירי בשפת C זהה לעבודה עם מערך רגיל מלבד האפשרות להשתמש בפונקציה המיוחדת `quick_insert` אשר מבצעת את ההכנסה המהירה למערך. הפונקציה מקבלת מערך אוורירי `a`, את גודלו `n`, את `x` - האיבר שרוצים להכניס, ואת `i` - האינדקס במערך שאליו רוצים להכניס את `x`. חתימתה של הפונקציה היא:

```
void quick_insert(int* a, int n, int x, int i);
```

בתום הקריאה, `a[i]` יכיל את הערך `x` והאיברים מהמקום ה-`i` ועד הסוף יוזזו. שימו לב כי הפונקציה מניחה כי ניתן לדרוס את הערך שנמצא במקום האחרון במערך (כלומר `a[n-1]`), וזאת לצורך הזזת האיברים ופינוי המקום המיועד לאיבר המוכנס. במלים אחרות, על המשתמש לדעת כי המידע הנמצא בתא האחרון במערך, אם ישנו מידע כזה, ילך לאיבוד.

דוגמה: בהנחה ש-`a` הינו מערך בגודל 10 המכיל את האיברים הבאים:

4	6	10	12	13	14	16	18	20	22
---	---	----	----	----	----	----	----	----	----

אחרי הקריאה `quick_insert(a, 10, 2, 0)` התוכן של `a` יהיה:

2	4	6	10	12	13	14	16	18	20
---	---	---	----	----	----	----	----	----	----

כמו כן כי ניתן לקרוא לפונקציה עם מערך חלקי ע"י שליחת ערך `n` קטן יותר מגודל המערך (אולם לא ניתן לקרוא לפונקציה עם ערך `n` שהוא גדול מגודל המערך בפועל). לדוגמא, לאחר הקריאה

`quick_insert(a, 6, 8, 3)` התוכן של `a` יהיה:

2	4	6	8	10	12	14	16	18	20
---	---	---	---	----	----	----	----	----	----

בשאלה זו נרצה לממש את הפונקציה `airy_sort` הממיינת מערך אוורירי ב- $O(n \log n)$ זמן וב- $O(1)$ זיכרון נוסף.

שימו לב, לשאלה זו שני סעיפים המופיעים בעמודים 11 ו 13.



- 10 -

דרישות סיבוכיות:

- מקום נוסף $O(1)$
- זמן $O(\log n)$

[illegible]



- 12 -



ממשו את הפונקציה `airy_sort` המקבלת מערך אוורירי `a` וגודלו `n`, וממיינת את `a` ב- $O(n \log n)$ זמן וב- $O(1)$ זיכרון נוסף.

הניחו כי הפונקציה `quick_insert` נתונה לכם. ניתן להשתמש בפונקציה מסעיף א גם אם לא מימשתם אותה.



שאלה 4 (20 נקודות)

מטריצת הדאמרד $H(N)$ היא מטריצה בגודל $N \times N$ המכילה ערכים בוליאניים בלבד (0 או 1). התכונה המאפיינת של מטריצה הדאמרד היא שבין כל שתי שורות יש בדיוק $N/2$ ערכים שונים. לדוגמא (שימו לב כי אלה אינן המטריצות היחידות האפשריות):

$$H(1) = (1), \quad H(2) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad H(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

באופן כללי, ניתן לקבל את $H(2N)$ על ידי שרשור 4 העותקים הבאים: $H(2N) = \begin{pmatrix} H(N) & H(N) \\ H(N) & !H(N) \end{pmatrix}$

כאשר הכוונה ב- $!H(N)$ הוא להיפוך כל הערכים הבוליאניים במטריצה $H(N)$.

ממשו את הפונקציה **הרקורסיבית** `createHadamard` שמקבלת מספר N ומחזירה מטריצת הדאמרד בגודל $N \times N$. הניחו ש- N הוא חזקה שלמה של 2. לרשותכם שתי פונקציות להקצאה ושחרור זיכרון של מטריצות:

```
int** allocateMatrix(int n)
```

מקצה זכרון למטריצה בגודל $n \times n$ ומחזירה מצביע לזכרון המוקצה. ניתן להשתמש בערך המוחזר `m` כאילו היה מטריצה שהוגדרה באופן רגיל (`int m[n][n]`).

```
void freeMatrix(int** m, int n)
```

משחררת זיכרון שהוקצה באמצעות `allocateMatrix`.

הקפידו לשחרר את כל הזיכרון שאתם מקצים, מלבד הזיכרון של המטריצה הסופית.



- 15 -

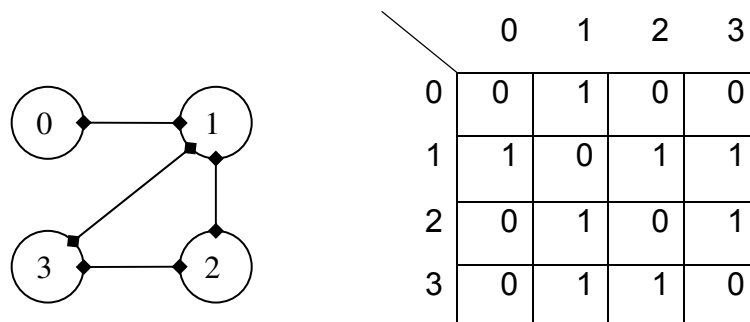


שאלה 5 (20 נקודות)

גרף מורכב מקבוצה של צמתים וקבוצה של קשתות, כאשר כל קשת מחברת שני צמתים. שני צמתים מחוברים ע"י קשת נקראים שכנים.

אחת מהדרכים לייצוג מבנה הגרף היא מטריצת סמיכויות. בשיטה זו, הגרף מיוצג כמטריצה בגודל $N \times N$ כאשר N הוא מספר הצמתים. התא i, j במטריצה מכיל "1" אם ישנה קשת בגרף קשת מהצומת של i לצומת של j , ו-"0" אחרת. שימו לב שמטריצת הסמיכויות הינה סימטרית. כמו כן אין משמעות לערכים באלכסון הראשי כיוון שהם מייצגים יחס סמיכות בין צומת לעצמה. לצרכי השאלה אתם יכולים להניח כי האלכסון הראשי מכיל אפסים.

דוגמה לייצוג גרף ע"י מטריצת סמיכויות:

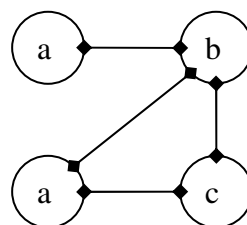


בשאלה זו מטרתנו תהייה לצבוע צמתי גרף בעל N צמתים תוך שימוש ב- k צבעים בלבד. צביעה תוגדר כחוקית אם כל צומת נצבע בצבע אחד ואין לו אף שכן שנצבע באותו הצבע. עליכם לממש את הפונקציה `color_graph` המקבלת את מטריצת הסמיכויות m המייצגת את הגרף ואת מספר הצבעים העומדים לרשותנו k ומדפיסה צביעה אחת אפשרית של הגרף. הנחות:

- מספר הצמתים N מוגדר ע"י `#define`.
- הצבעים האפשריים יקודדו ע"י אותיות אנגליות קטנות לפי הסדר `a, b, c, d, ...`. ניתן להניח כי מספר הצבעים k יהיה קטן ממספר האותיות בא"ב האנגלי.
- הינכם רשאים להשתמש בפונקציית הספרייה `printf` לצורך ההדפסה.
- על הפונקציה לעבוד בשיטת ה-**backtracking**.

אם ניתן לצבוע את הגרף ע"י k צבעים אז יש להדפיס את קודי הצבעים בשורה אחת, לפי סדר הצמתים. אחרת, יש להדפיס את ההודעה "Not colorable".

לדוגמא, הגרף לעיל ניתן לצביעה ע"י 3 צבעים. צביעה אפשרית היא: a b c a



אולם, לא קיימת צביעה חוקית המשתמשת בשני צבעים בלבד.



- 17 -



- 18 -



- 19 -



- 20 -



- 21 -