



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

סמסטר קיץ תשע"ג

מבחן סופי מועד ב', 29 באוקטובר 2013 (*)

(*) המבחן משמש גם עבור מועד ג' סמסטר אביב תשע"ג.

מספר סטודנט:

3	0	3	0	1	6	9	5	0
---	---	---	---	---	---	---	---	---

 רשום/ה לקורס:

2	3	4	1	1	7
---	---	---	---	---	---

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 22 עמודים 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. ניתן בהחלט להשתמש בעיפרון ומחקק, 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.
- אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה, למעט פונקציות קלט/פלט והקצאת זיכרון (malloc). ניתן להשתמש בטיפוס bool המוגדר ב-stdbool.h.
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- ניתן להשתמש בהקצאות זכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זכרון.
- נוסחאות שימושיות:

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

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

צוות הקורס 234114/7

מרצה: ד"ר יחיאל קמחי.

מתרגל: ישראל גוטר.

בהצלחה!



שאלה 1 (25 נקודות):

א. (10 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה f המוגדרת בקטע הקוד הבא, כפונקציה של n . אין צורך לפרט שיקולים.

```
int f(int n)
{
    int i, k, s;
    int sum=0;

    for (i=1; i<=n; i+=i)
    {
        for (k=1; k<=i; k+=2)
        {
            s=i;
            while (s>=1)
            {
                s/=4;
                sum+=s;
            }
        }
    }
    return sum;
}
```

סיבוכיות מקום: $\Theta(1)$

אמור להיות:
 $\Theta(n \cdot \log(n))$

סיבוכיות זמן:
הסבר משה

ב. (15 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה f המוגדרת בקטע הקוד הבא, כפונקציה של n . אין צורך לפרט שיקולים. בנוסף, מצאו מהי תוצאת הפונקציה במקרים המפורטים

```
int f(int a[], int n)
{
    int *p;
    if (n==0) return 0;
    if (n==1) return *a;
    p=a+n/2;
    return *p+f(a,p-a)+f(p+1,a+n-p-1);
}
```

סיבוכיות מקום: $\Theta(\log n)$

1. סיבוכיות זמן: $\Theta(n)$

2. מה תהיה תוצאת הפונקציה על המערכים הבאים:

א. 1 2 3 4 5 6

ב. -2 2

21

0



שאלה 2 (30 נקודות):

כתבו פונקציה `int find (int a[], int n1, int b[], int n2)` הפונקציה מקבלת כפרמטרים 2 מערכים לא ריקים של מספרים שלמים, `a` ו-`b`. `n1` הוא אורכו של המערך `a`. `n2` הוא אורכו של המערך `b`. המערכים ממיינים מהקטן לגדול, משמאל לימין. הפונקציה מחזירה כתוצאה את ההפרש המינימלי בין 2 מספרים כלשהם מבין כלל המספרים שמופיעים בשני המערכים.

דוגמה:

בהינתן המערכים

-15 -7 31 67 79
-10 24 27 70 95 150

ההפרש המינימלי הוא 3, ובדוגמא זו הוא קיים בין 2 זוגות של מספרים:

-10 -7

24 27

שימו לב ש-2 המספרים בזוג אחד יכולים להימצא שניהם באותו המערך, וגם יכולים להימצא, כל אחד, במערך אחר.

דרישות:

1. אסור לשנות את איברי 2 המערכים.
2. סיבוכיות מקום קבועה $\Theta(1)$.
3. סיבוכיות זמן מיטבית.

ציינו מהי סיבוכיות הזמן (במקרה הגרוע) של הקוד שכתבתם:

מהי הסיבוכיות?

$\Theta(\quad)$

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

```
{
```

```
    int min = findmindiff(a, n1);
```

```
    if (min == 0) return 0;
```

```
    if (n2 > 1)
```

```
        int min2 = findmindiff(b, n2)
```

```
        if (min2 == 0) return 0;
```




```
int find (int a[], int n1, int b[], int n2)
{
    int min = (a[0] > b[0] ? a[0] - b[0] : b[0] - a[0]);
    if (min == 0) return 0;
    if (n1 > 1)
    {
        int min1 = findmindiff(a, n1);
        if (min1 == 0) return 0;
        if (min1 < min) min = min1;
    }
    if (n2 > 1)
    {
        int min2 = findmindiff(b, n2);
        if (min2 == 0) return 0;
        if (min2 < min) min = min2;
    }
}
```

המערך הבא (6)



המאתב
מאתב

```
for(int i=0, int j=0 ; i+j < n-1 ; )  
{
```

תנאי עצירה לא נכון!

```
    int diff;
```

```
    if (a[i] > b[j])
```

```
    {
```

```
        diff = a[i] - b[j];
```

```
        if (diff < min) min = diff;
```

```
        j++;
```

```
    }
```

```
    else if (a[i] < b[j])
```

```
    {
```

```
        diff = b[j] - a[i];
```

```
        if (diff < min) min = diff;
```

```
        i++;
```

```
    }
```

```
    else return 0;
```

```
    }  
    return min;
```

```
int findminDiff(int arr[], int n)
```

```
{    int min = arr[1] - arr[0];
```

```
    for (int i = 1 ; i < n-1 ; i++)
```

```
    {
```

```
        int diff = arr[i+1] - arr[i];
```

```
        if (diff < min) min = diff;
```

```
        if (min == 0) return 0;
```

```
    }
```

```
    return min;
```

```
}
```




שאלה 3 (20 נקודות):

- נתון מערך דו-ממדי בגודל $N \times M$ (הנתונים ע"י `#define`) וערכיו מספרים שלמים (יכולים להיות גם שליליים). כתבו פונקציה `int min_cost(int m[N][M])` המוצאת מסלול שמחירו מזערי בתנאים:
- המסלול מתחיל באיבר `m[0][0]` ומסתיים באיבר `m[N-1][M-1]`.
 - בכל צעד המסלול מתקדם רק לכיוונים ימינה או למטה (הוא מכיל בדיוק $N+M-1$ איברים).
 - מחיר המסלול הוא סכום הערכים בתאים (משבצות) בהם הוא עובר.
 - הפונקציה מחזירה את מחיר המסלול ה"זול" ביותר שמצאה.

1	9	9	9	9	9	9	9	9
1	9	9	9	9	-9	-9	9	9
1	9	9	1	9	9	9	8	9
1	1	1	1	1	9	1	0	9
1	9	1	9	9	9	2	8	-9
8	8	1	1	1	1	1	1	1
2	1	1	9	1	1	8	8	8
0	0	9	1	1	1	1	1	1

במערך שבדוגמה משמאל יש שלשה מסלולים שונים שמחירם מזערי (16) – אחד מהם מודגש – וכולם מכילים 1-ים בלבד (בחירת הערכים "כולם 1" נעשתה רק כדי לאפשר זיהוי קל של המסלול).

ממשו את הפונקציה `min_cost(int m[N][M])` לפי התנאים שלעיל בסיבוכיות זמן $O(N \times M)$ ובסיבוכיות מקום $O(M)$ (למעשה $O(\min(N, M))$).
בנוסף 3 נקודות למי שמוכיח מדוע סיבוכיות הזמן היא $\Omega(N \times M)$ (כלומר, לא פחות מ- $N \times M$).

שימו לב: הפתרון אינו דורש שימוש ב- `backtracking` ("ניסוי ותעיה")

הערה: אם אינכם מצליחים לפתור בדרישות הסיבוכיות האמורות, תוכלו לפתור בסיבוכיות שאינה מקיימת את הדרישות. במקרה זה חובה לציין מהי סיבוכיות הזמן של הפתרון שנתתם, ומהי סיבוכיות המקום. פתרון כזה יוכל לקבל ניקוד חלקי בהתאם למאפייניו.

סיבוכיות זמן: $\Theta(\quad)$ סיבוכיות מקום: $\Theta(\quad)$

```
int min_cost(int m[N][M])
```



שאלה 4 (25 נקודות) :

- בשאלה זו עליכם לרצף רצפה בגודל $N \times M$ באמצעות אוסף שמכיל בדיוק L אריחים מלבניים. גדלי האריחים מוגדרים במערך הדו-מימדי $\text{tiles}[L][2]$. מימדי האריח ה- i הם $\text{tiles}[i][0] \times \text{tiles}[i][1]$. מניחים שהפרמטרים N, M, L , מוגדרים ע"י `#define`.
1. מובטח שהשטח הכולל של כל L האריחים ביחד הוא $N \times M$ (כל אריח מופיע פעם יחידה).
 2. לכן: כדי לרצף את הרצפה האמורה חובה להשתמש בכל האריחים, כל אריח – פעם יחידה, ואסור לשני אריחים שמונחים על הרצפה לחפוף ביניהם.
 3. כל אריח ניתן לסובב ב-90 מעלות. למשל, אריח בגודל 3×4 ניתן לפרש כ"גובה 3 ואורך 4" או כ"גובה 4 ואורך 3".
- עליכם לממש פונקציה `int tile_floor(int tiles[L][2])` אשר מחזירה 1 אם ניתן לרצף את הרצפה באופן מושלם (כל נקודה ברצפה מכוסה) ו-0 אחרת. יש להשתמש ב-`backtracking` כפי שנלמד בכיתה.

```
int tile_floor(int tiles[L][2])
```