



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

סמסטר חורף תשע"ג

מבחן מסכם מועד ב', 2 אפריל 2013

--	--	--	--	--	--	--	--	--	--

מספר סטודנט:

משך המבחן: 2.5 שעות.
חומר עזר: אין להשתמש בכל חומר עזר.

הנחיות כלליות:

- מלאו את הפרטים בראש דף זה ובדף השער המצורף, בעט בלבד.
- בדקו שיש 18 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 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.

הנחיות תכנות כלליות, אלא אם מצויין אחרת בשאלה:

- אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה, למעט פונקציות קלט/פלט והקצאת זיכרון (malloc).
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- ניתן להשתמש בהקצאות זכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זכרון.

צוות הקורס 234114/7

מרצים: ד"ר תמיר לוי, ד"ר רן רובינשטיין, פרופ' רון קימל (מרצה אחראי).

מתרגלים: תהילה מייזלס, שי גרץ, סינטיה דיזנפלד, נחשון כהן, נדיה לבאי, ורד כהן, תומר ארזי, יעל ארז, אסף ישראל (מתרגל אחראי).

בהצלחה!



- 2 -



שאלה 1 (25 נקודות)

בכל אחד מקטעי הקוד הבאים, חשבו את סיבוכיות הזמן והמקום של הפונקציה func כתלות ב-n. יש לכתוב את הביטוי הפשוט ביותר לסיבוכיות בכל המקרים.

א.

```
int func(int n)
{
    if (n < 1)
        return 0;

    int x = 1;
    while (x*x < n) {
        x++;
    }

    return x + func(n/2);
}
```

סיבוכיות זמן: $\Theta(\sqrt{n})$ סיבוכיות מקום: $\Theta(\log(n))$

ב.

```
void func(int n)
{
    int x = 0;
    for (int k=0; k<n; ++k) {
        x += 2*k;
    }

    while (x>0) {
        x /= 2;
    }
}
```

סיבוכיות זמן: $\Theta(n)$ סיבוכיות מקום: $\Theta(1)$



- 4 -



ג.

```
int func(int n)
{
    int k=5;

    if (n<=1) {
        return n;
    }

    int sum = 0;
    for (int i=0; i<k; ++i) {
        sum += func(n/k);
    }

    return sum;
}
```

סיבוכיות זמן: $\Theta(n)$ סיבוכיות מקום: $\Theta(\log(n))$



- 6 -



שאלה 2 (25 נקודות)

סעיף א

בסעיף זה נתונה קבוצת מספרים, כולם בטווח 0 עד $k-1$. נזכיר שההיסטוגרמה $hist[]$ של קבוצה זו היא מערך באורך k , המכיל בתא $hist[i]$ את כמות המספרים בקבוצה ששווים ל- i .

ממשו פונקציה המקבלת את ההיסטוגרמה $hist[]$ ואת אורכה k , ומחזירה את ממוצע המספרים בקבוצה. לדוגמה, עבור ההיסטוגרמה הבאה: $hist[3] = \{3, 0, 1\}$, הקבוצה מכילה את הערכים 0,0,0,2 ולכן הפונקציה תחזיר 0.5.

דרישות סיבוכיות: על הפונקציה לעבוד בסיבוכיות זמן $O(k)$ וסיבוכיות מקום $O(1)$.

```
double average(int hist[], int k) {  
    int count = 0, sum = 0;  
    for (int i = 0; i < k; i++) {  
        count += hist[i];  
        sum += i*hist[i];  
    }  
    return count ? (double)sum / count : 0.0;  
}
```



- 8 -



סעיף ב

(1) נתון מערך $x[]$ באורך n , ונדרש לחשב את המספר y המביא למינימום את הפונקציה הבאה:

$$\sum_{i=0}^{n-1} (y - x[i])^2$$

מהו הערך y המביא למינימום את הפונקציה הנ"ל? כתבו במילים או כנוסחה:

$$y = \frac{\sum_{i=0}^{n-1} x[i]}{n} = X \text{ ממוצע ערכי } X$$

(2) עתה, נניח כי בנוסף למערך $x[]$ נתון לנו גם מערך של משקולות $w[]$ באורך n , כאשר $w[i]$ הוא המשקל של האיבר $x[i]$. ברצוננו לחשב את המספר y המביא למינימום את הפונקציה הבאה:

$$\sum_{i=0}^{n-1} w[i] \cdot (y - x[i])^2$$

מהו הערך y המביא למינימום את הפונקציה החדשה? חשבו וכתבו את התוצאה למטה, במילים או כנוסחה. באמצעות חישוב זה, ממשו את הפונקציה (בעמוד הבא) המקבלת את המערך $x[]$, את אורכו n ואת המערך $w[]$, ומחזירה את y .

דרישות סיבוכיות: יש לעמוד בסיבוכיות זמן $O(n)$ ומקום $O(1)$.

$$y = \frac{\sum_{i=0}^{n-1} w[i] x[i]}{\sum_{i=0}^{n-1} w[i]} = X \text{ ממוצע משוקלל של ערכי } X$$

(3) לבסוף, נניח שברצוננו להביא למינימום פונקציה שלישית, הנתונה על ידי:

$$\sum_{i=0}^{n-1} f(y, x[i]) \cdot (y - x[i])^2$$

מהו הערך y המביא פונקציה זו למינימום עבור הבחירה הבאה של f ?

$$f(y, x[i]) = \frac{1}{|y - x[i]|}$$

חשבו וכתבו את התוצאה במילים או כנוסחה:

$$y = X \text{ החציון של ערכי } X$$



- 10 -



מימוש הפונקציה עבור סעיף ב' (2):

```
double compute_y(double x[], int n, double w[]) {  
    double weightedSum = 0, sumWeights = 0;  
    for (int i = 0; i < n; i++) {  
        weightedSum += x[i]*w[i];  
        sumWeights += w[i];  
    }  
    return sumWeights ? weightedSum / sumWeights : 0;  
}
```



- 12 -



שאלה 3 (25 נקודות)

נתון מערך חד מימדי $a[]$ באורך n המכיל ערכי 0 ו-1 בלבד. לדוגמה:

$a[] = [0,1,1,1,1,1,1,0,1,1,1,1,0,1,1]$

עליכם לכתוב תוכנית המקבלת כקלט את המערך a ואת אורכו n , ומחליפה כל ערך במערך שאינו 0 במרחק ממנו לנקודת האפס הקרובה ביותר (מימין או משמאל). שימו לב שב"מרחק" הכוונה פשוט למספר הצעדים במערך שיש לבצע על מנת להגיע לאפס הקרוב ביותר. לדוגמה, עבור המערך a הנ"ל, הפלט יהיה:

$a[] = [0,1,2,3,3,2,1,0,1,2,3,2,1,0,1,2]$

הערות:

- על הפתרון לעמוד בסיבוכיות זמן $O(n)$ כאשר n הוא אורך המערך, וסיבוכיות מקום $O(1)$.
- ניתן להניח שהמערך מכיל לפחות 0 אחד (הוא אינו כולו 1).

```
void zero_dist(int a[], int n) {  
    int dist = n;  
    for (int i = 0; i < n; ++i) {  
        if (a[i] == 0)  
            dist = 0;  
        a[i] = dist;  
        dist++;  
    }  
    dist = n;  
    for (int i = n-1; i >= 0; --i) {  
        if (a[i] == 0)  
            dist = 0;  
        a[i] = (a[i] < dist) ? a[i] : dist;  
        dist++;  
    }  
}
```

הסבר: נבצע שתי סריקות, אחת מכל צד. בראשונה נאתחל כל תא להיות המרחק מהאפס השמאלי ביותר. שימו לב כי עד האפס הראשון כל הערכים גדולים/שווים ל- n . בסריקה השניה נדאג שהערך המינימאלי מבין המרחק לאפס שמאלי (תוכן התא) והימני יכנס לכל תא במערך. כיוון, שהערכים בתחילת המערך (לפני האפס הראשון) גדולים ממש מ- n הרי שכולם יוחלפו במרחק לאפס השמאלי הראשון.



- 14 -



שאלה 4 (25 נקודות)

בשאלה זו נתונים n חפצים, וכן מערך $weight[]$ באורך n המכיל בתא i -ה את משקלו של החפץ i -ה. ברשותנו משאית איתה היינו רוצים להוביל את החפצים. אולם, על מנת לצאת לדרך, הוחלט שיש להעמיס על המשאית משקל כולל שהוא לפחות w_1 , ולכל היותר w_2 , כאשר w_1 ו- w_2 ערכים נתונים. ממשו את הפונקציה הבאה, המקבלת את המערך $weight[]$, את גודלו n , ואת הערכים w_1 ו- w_2 , ומחזירה את המספר המקסימאלי של חפצים שניתן להעמיס על המשאית ואשר סכום משקליותיהם בטווח הנתון. במידה ואין אף פתרון חוקי, יש להחזיר -1. כמו כן, במידה ומוחזר פתרון יש לכתוב אותו לתוך המערך $sol[]$, כאשר $sol[i]$ מכיל 1 אם החפץ i -ה כלול בפתרון, ו-0 אחרת.

הערות:

א. על הפתרון להיות רקורסיבי ולהשתמש ב-backtracking.

ב. ניתן להגדיר פונקציות עזר כרצונכם.

```
int load(int weight[], int n, int w1, int w2, int sol[]) {
    if (w2 < 0)
        return -1;
    if (n == 0)
        return (w1 <= 0 ? 0 : -1);
    sol[0]=0;
    int res1 = load(weight+1,n-1,w1,w2,sol+1);
    int solTmp[n];
    solTmp[0] = 1;
    int res2 = load(weight+1,n-1,w1-weight[0], w2-weight[0],
                    solTmp+1);
    if (res2 != -1 && res2 >= res1) {
        for (int i = 0 ; i < n ; i++)
            sol[i] = solTmp[i];
        return res2+1;
    }
    return res1;
}
```

הטכניון, מכון טכנולוגי לישראל מבוא למדעי המחשב מ'ח'

הסבר: הרקורסיה מתבצעת על מערך החפצים. בכל צעד נבצע שתי קריאות רקורסיביות: האחת לאחר שהפריט הנוכחי הוכנס למשאית, והשניה ללא הפריט. נבחר בפיתרון הטוב מהשניים ואותו נחזיר. יש לשים לב, שבשביל אחת הקריאות יש להקצאות מערך עזר כדי לשחזר את הפיתרון במידה ונבחר בה. לאחר כל השמה של פריט נעדכן את החסמים כלפי מטה, ולבסוף נבדוק האם החסם העליון חיובי והתחתון שלילי.

הבעיה דומה מאוד לבעיית הצריבה שראינו בתרגולים למעט התנאי הנוסף של החסם התחתון.

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.



- 17 -



- 18 -