



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

### סמסטר חורף תשע"ד

### מבחן מסכם מועד ב', 27 מרץ 2014

<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 (מודפס או ממוחשב).

#### 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 תיבדקנה.
- יש לכתוב באופן ברור, נקי ומסודר. ניתן בהחלט להשתמש בעיפרון ומחק, פרט לדף השער אותו יש למלא בעט.
- בכל השאלות, הנכם רשאים להגדיר ולממש פונקציות עזר כרצונכם. לנוחיותכם, אין חשיבות לסדר מימוש הפונקציות בשאלה, ובפרט ניתן לממש פונקציה לאחר השימוש בה.
- בשאלה 4 בלבד, ניתן לשאיר פתרון ריק עבור 20% מהציון.

#### הנחיות תכנות כלליות, אלא אם מצוין אחרת בשאלה:

- אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה, למעט פונקציות קלט/פלט והקצאת זיכרון (`malloc`), אלא אם נכתב אחרת.
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- ניתן להשתמש בהקצאות זיכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זיכרון.
- ניתן להשתמש בטיפוס `bool` המוגדר ב-`stdbool.h`.

צוות הקורס 234114/7

**מרצים:** גב' אנסטסיה דוברובינה, ד"ר רן רובינשטיין, פרופ' רון קימל (מרצה אחראי).

**מתרגלים:** נמרוד סבן פרטוש, אריק יודין, נדיה לבאי, אביב סגל, בת-חן גולדן, רן ברנשטיין, דור הריס, ברק פת, נחשון כהן (מתרגל אחראי).

**בהצלחה!**



- 2 -



## שאלה 1 (25 נקודות)

### סעיף א

נתונות הפונקציות func1(), func2(), func3(). כתבו מתחת לכל פונקציה מה היא מדפיסה:

```
int f(int n)
{
    int x=0;
    while (x*x < n) {
        x++;
    }
    return x*x;
}

void func1()
{
    printf("%d\n", f(8));
    printf("%d\n", f(f(f(8))));
    printf("%d\n", f(-8));
}
```

פלט של func1():

9
9
0



- 4 -



```
void g(int* p, int* q)
{
    *(p + *q) = 2*(*q);
    q++;
}

void func2()
{
    int a[6];
    for (int i=0; i<6; i++) {
        g(a,&i);
    }
    for (int i=0; i<6; i++) {
        printf("%d\n", a[i]);
    }
}
```

פלט של func2():

0

2

4

6

8

10



- 6 -



```
void swap(char* p, char* q)
{
    char tmp = *p;
    *p = *q;
    *q = tmp;
}

int h(char* s, int n)
{
    if (s[n]==0) {
        return 0;
    }
    if (s[n+1]==0) {
        return 1;
    }
    int k = h(s+1,n+1);
    swap(s,s+k+1);
    return k+2;
}

void func3()
{
    char s[] = "intro to cs";
    int x = h(s,0);
    printf("x = %d\n", x);
    printf("s = %s\n", s);
}
```

פלט של func3():

```
x = 11
s = sc ot ortni
```



- 8 -





## סעיף ב

עבור כל אחת מהפונקציות הבאות  $\text{func4}()$ ,  $\text{func5}()$ , כתבו את סיבוכיות הזמן והמקום שלהן כתלות ב- $n$  ו- $m$  במקום המתאים. כמו כן, תארו בקצרה (משפט אחד) מה מחזירה הפונקציה  $\text{func4}()$ .

```
int func4(int a[], int n, int m)
{
    int h[m];
    for (int i=0; i<m; ++i) {
        h[i] = 0;
    }
    for (int i=0; i<n; ++i) {
        h[a[i]]++; // הניחו שאברי המערך בתחום 0..m-1
    }
    int x=0;
    for (int i=0; i<m; ++i) {
        x += i*h[i];
    }
    return x;
}
```

סיבוכיות זמן:  $\Theta(m+n)$  סיבוכיות מקום:  $\Theta(m)$

מהו ערך ההחזרה של הפונקציה  $\text{func4}()$  ?

סכום איברי המערך

```
void func5(int a[], int n, int m, int b[])
{
    if (n==0) {
        return;
    }
    *b = func4(a,n,m);
    func5(a+1,n-1,m,b+1);
}
```

סיבוכיות זמן:  $\Theta(n^2+m^2)$ , סיבוכיות מקום:  $\Theta(m+n)$



- 10 -



## שאלה 2 (25 נקודות)

### סעיף א

נתון מערך מטיפוס double, ממין בסדר עולה. עליכם לממש פונקציה המקבלת את המערך  $a[]$ , את גודלו  $n$ , וערך נוסף  $x$ , ומחזירה את אינדקס האיבר מ- $a$  שקרוב ביותר בערכו ל- $x$ . לדוגמה, עבור המערך הבא:

1.3	5.2	6.0	7.1	9.8	12.3	14.5
-----	-----	-----	-----	-----	------	------

ועבור  $x=4.5$ , הפונקציה תחזיר 1 כיוון ש- $a[1]$  הינו הערך הקרוב ביותר ל- $x$  במערך.

### הערות:

- ניתן להניח שאין איברים החוזרים על עצמם במערך  $a$ .
- במידה ויש שני מספרים במרחק זהה ל- $x$ , יש להחזיר אחד מהם לבחירתכם.
- מומלץ לממש פונקציית עזר המחשבת את המרחק (בערך מוחלט) בין שני מספרים.
- על הפונקציה לעמוד בסיבוכיות זמן  $O(\log(n))$  וסיבוכיות מקום נוסף  $O(1)$ .

```
int closest(double a[], int n, double x) {  
  
    int high=n-1, low=0;  
    //corner cases. Not needed here, but safer.  
    if(x<a[low]) return low;  
    if(x>a[high]) return high;  
    //binary search for x.  
    while(low<high-1) {  
        mid=(high+low)/2;  
        if(a[mid]==x)  
            return mid;  
        if(a[mid]<x)  
            low=mid;  
        else  
            High=mid;  
    }  
    //now x is between a[low] and a[high]  
    If(distance(a[low], x)<distance(a[high], x))  
        Return low;  
    Return high;  
}
```



```
Double distance(double a, double b) {  
    int abs=a-b;  
    if(abs<0)  
        return (-1)*abs;  
    return abs;  
}
```



## סעיף ב

ממשו פונקציה המקבלת מערך  $a[]$ , את גודלו  $n$ , ושני מספרים  $\min$  ו- $\max$  (כאשר  $\max > \min$ ), ומחזירה את מספר האיברים  $a[i]$  במערך המקיימים  $\max > a[i] > \min$ . למשל, עבור המערך מהסעיף הקודם ועבור  $\max=10$ ,  $\min=5$ , הפונקציה תחזיר 4 (כיוון שהאיברים 5.2, 6.0, 7.1, 9.8 הם בתחום בין  $\min$  ל- $\max$ ).

## הערות:

- ניתן להניח שאין איברים החוזרים על עצמם במערך  $a$ .
- ניתן להניח שאין במערך  $a$  איברים השווים ל- $\min$  או  $\max$ .
- בסעיף זה ניתן להשתמש בפונקציה מהסעיף הקודם, גם אם לא מימשתם אותה.
- על הפונקציה לעמוד בסיבוכיות זמן  $O(\log(n))$  וסיבוכיות מקום נוסף  $O(1)$ .

```
int range(double a[], int n, double min, double max) {  
    int closest_to_min=closest(a, n, min);  
    closest_to_min+=(a[closest_to_min]<min);  
    //now closest_to_min points to the first value  
    //greater than min  
    int closest_to_max=closest(a, n, max);  
    closest_to_max-=(a[closest_to_max]>max);  
    return closest_to_max-closest_to_min+1;  
}
```



### שאלה 3 (25 נקודות)

#### סעיף א

בסעיף זה נתון מערך  $s[]$  של מחרוזות, כאשר ידוע שהמחרוזות בעלות אורך שונה זה מזה. עליכם לממש פונקציה (בעמוד הבא) המקבלת את המערך  $s$  ואת אורכו  $n$ , ומחזירה את האינדקס של המחרוזת השנייה הקצרה ביותר במערך, כלומר, המחרוזת שהייתה מופיעה בתא השני במערך לו היינו ממיינים אותו בסדר עולה של אורך המחרוזת. למשל, עבור המערך הבא:

```
char* s[] = { "abcde", "a", "bc", "234114" };
```

הפונקציה תחזיר 2, כיוון ש- $s[2]$  ("bc") הינה המחרוזת השנייה הקצרה ביותר במערך (אחרי המחרוזת "a" שהיא הקצרה ביותר).

#### הערות:

1. לצורך השאלה נתונה לכם פונקציית העזר:

```
int strlen(char* str);
```

המחזירה את אורכה של המחרוזת  $str$ .

2. ניתן להניח שיש לפחות שתי מחרוזות במערך.

#### דרישות סיבוכיות:

- **מקום:** על הפתרון לעמוד בסיבוכיות מקום  $O(1)$ .
- **זמן:** הניחו שפעולת חישוב של אורך מחרוזת הינה פעולה "כבדה" שיש לצמצם בביצועה. לפיכך, בפתרון השאלה מותר להפעיל את הפונקציה  $strlen()$  לכל היותר פעם אחת עבור כל מחרוזת במערך (וסה"כ  $n$  פעמים לכל היותר). למען הסר ספק, כל פעולת חישוב של אורך מחרוזת, גם ללא קריאה ל- $strlen$ , תחשב כקריאה ל- $strlen$ .

- 15 -



## סעיף ב

בסעיף זה נתון מערך  $s[]$  של מחרוזות, כאשר ידוע שכל המחרוזות בו בעלות אורך של לכל היותר  $m$ . שימו לב שבסעיף זה ייתכנו כמה מחרוזות בעלות אותו האורך במערך  $s$ . עליכם לממש פונקציה המקבלת את המערך  $s$ , את אורכו  $n$ , וכן את אורך המחרוזת המקסימאלית  $m$ , וממיינת את  $s$  בסדר עולה לפי אורך המחרוזת. למשל, עבור המערך הבא:

```
char* s[] = { "abcde", "aa", "bc", "234114" };
```

הפונקציה עשויה למיין מחדש את  $s$  באופן הבא:

```
char* s[] = { "aa", "bc", "abcde", "234114" };
```

שימו לב שאין חשיבות לסדר בין מחרוזות שאורכן זהה.

**דרישות סיבוכיות:** על הפונקציה לעמוד בסיבוכיות זמן  $O(n \cdot m)$  וסיבוכיות מקום  $O(n+m)$ .

הערה: בסעיף זה ניתן להשתמש בפונקציה  $\text{strlen}()$  מהסעיף הקודם.

```
void sort(char* s[], int n, int m) {  
    char *helper[n];  
    int lengths[n], curr=0; //next to fill entry  
    for(int i=0; i<n; ++i){  
        helper[i]=s[i];  
        lengths[i]=strlen(s[i]);  
    }  
    for(int k=0; k<=m; ++k) //for each length  
        for(int i=0; i<n; ++i) { //copy strings of length k to s  
            if(lengths[i]==k)  
                s[curr++]=helper[i];  
        }  
}
```





- 17 -



### שאלה 4 (25 נקודות)

ממשו פונקציה המקבלת מערך של שלמים  $a[]$ , את גודלו  $n$ , וערך שלם  $diff$ , ובודקת אם ניתן לחלק את איברי המערך לשתי קבוצות כך שההפרש בין סכומיהן יהיה  $diff$ . הפונקציה מחזירה  $true$  אם קיימת חלוקה כזו, או  $false$  אחרת. לדוגמה, עבור המערך:

6	1	2	4	8	5	3
---	---	---	---	---	---	---

ועבור ההפרש  $diff=1$ , הפונקציה תחזיר  $true$  כיוון שניתן לחלק את המערך לשתי קבוצות באופן הבא:

$$6+1+8 = 15 \quad 2+4+5+3 = 14$$

הערות:

- שימו לב שכל איבר ב- $a$  חייב להיות שייך לאחת משתי הקבוצות.
- יש לפתור את השאלה ברקורסיה.

`/*explanation:`

`At each step we pick  $a[0]$  to be either in group A or group B.`

`Diff is the difference between the groups + the user sent diff.`

`After finishing all entries, diff should be zero if the  
difference between the groups is diff`

`Result is true if either placing  $a[0]$  at group A or placing it at  
group B return true;`

`Note: if the first return true we do not compute the second  
because of lazy evaluation.`

`*/`

`bool split(int a[], int n, int diff) {`

`if(n==0)`

`return diff==0;`

`return split(a+1, n-1, diff-a[0]) ||`

`split(a+1, n-1, diff+a[0]);`

`}`