



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

סמסטר אביב תשס"ה

### פתרון מבחן מסכם מועד ב', 20 ספטמבר 2005

<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: 2.5 שעות.  
חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני.

#### הנחיות והוראות:

- מלאו את הפרטים בראש דף זה.
- בדקו שיש 10 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 תבדקנה.
- 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 בשאלה.

צוות הקורס 234114
<b>מרצים:</b> איתן אביאור, דר' רועי פרידמן (מרצה אחראי).
<b>מתרגלים:</b> רן רובינשטיין, מיכל הולצמן-גזית.

צוות הקורס 234117
<b>מרצים:</b> רועי מלמד, דר' רועי פרידמן (מרצה אחראי).
<b>מתרגלים:</b> עידו פלדמן, גיא פליישר, אייל רוזנברג, אולג רוכלנקו.

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

**בהצלחה!**



**שאלה 1 (20 נקודות)**

סעיף א (8 נקודות)

נתונה הפונקציה הרקורסיבית הבאה:

```
int f(int n)
{
    if (n <= 1) return 1;
    return n + f(sqrt(n));
}
```

שכתבו את הפונקציה כך שתבצע את אותה הפעולה, אך ללא שימוש ברקורסיה:

```
int g(int n) {
    int sum = 1;
    while (n>1) {
        sum += n;
        n = sqrt(n);
    }
    return sum;
}
```



סעיף ב (12 נקודות)

1. כתבו את סיבוכיות הזמן והמקום של הפונקציה  $\text{strange}(n)$  כפונקציה של  $n$  (כאשר  $n \geq 1$  טבעי):

```
void strange(int n)
{
    int i = 0;

    while (i < n/2)
    {
        aux(i);
        aux(n-i);
        i += 2;
    }

    return;
}
```

```
void aux(int m)
{
    int j, k;

    for (j=0, k=1; j < m; j++)
        k *= 3;

    while (k)
        k /= 2;

    return;
}
```

סיבוכיות מקום:  $\Theta(\underline{\hspace{1cm}1\hspace{1cm}})$

סיבוכיות זמן:  $\Theta(\underline{\hspace{1cm}n^2\hspace{1cm}})$

2. כתבו את סיבוכיות הזמן והמקום של הפונקציה  $\text{weird}(n)$  כפונקציה של  $n$  (כאשר  $n \geq 1$  טבעי):

```
void weird(int n)
{
    int i;

    for (i = 1; i < n; i += 3) {
        helper(i);
    }

    return;
}
```

```
void helper(int m)
{
    int j;

    if (m <= 3)
        return;

    helper(m/2);

    for (j = 0; j < m; j++)
        printf("?");

    return;
}
```

סיבוכיות מקום:  $\Theta(\underline{\hspace{1cm}\log(n)\hspace{1cm}})$

סיבוכיות זמן:  $\Theta(\underline{\hspace{1cm}n^2\hspace{1cm}})$



## שאלה 2 (20 נקודות)

בשאלה זו נממש פונקציה המחשבת את ערכם של **ביטויים לוגיים**. ביטוי לוגי יכול לקבל את הערך 1 (אמת) או 0 (שקר), ומוגדר כאחד מהשניים הבאים:

- אחד המספרים החד-ספרתיים 0 או 1.
- צירוף של שני ביטויים לוגיים, עם פעולה לוגית ביניהם, מוקף בסוגריים וללא רווחים, באופן הבא:  
(ביטוי לוגי <פעולה> ביטוי לוגי). הפעולות הלוגיות האפשריות הן '&' ('וגם') ו-'|' ('או').

להלן מספר דוגמאות לביטויים לוגיים, וערכם:

ביטוי לוגי	ערך
0	0
(0 1)	1
((1&1) 0)	1
((1&(1 0))&(1&0))	0
(1&0&1)	ביטוי לא חוקי
((0))	ביטוי לא חוקי

ביטוי לוגי ייוצג בשאלה זו על ידי מערך של `char` (שימו לב: זהו מערך רגיל ולא מחרוזת, כיוון שהוא אינו מסתיים בתו 0). השלימו את הפונקציה הבאה, שמקבלת ביטוי לוגי `a[]` ואת אורך המערך `n` ומחזירה את ערך הביטוי כ-`int` (ניתן להניח כי הביטוי הלוגי שהפונקציה מקבלת חוקי) **על הפתרון להיות רקורסיבי** ולעבוד בסיבוכיות זמן  $O(n)$ .

```
int calc(char a[], int n) {
    int count = 0, i = 1;

    if (n==1) return a[0]-'0';
    do {
        if (a[i]=='(') count++;
        if (a[i]==')') count--;
        i++;
    } while (count>0);

    if (a[i]=='&') return calc(a+1,i-1) && calc(a+i+1,n-i-2);
    else return calc(a+1,i-1) || calc(a+i+1,n-i-2);
}
```



### שאלה 3 (20 נקודות)

נתונה מחרוזת  $s$  המורכבת מאותיות אנגליות קטנות בלבד וללא רווחים. בשאלה זו עליכם לכתוב פונקציה המקבלת את המחרוזת  $s$  ו**מתמצתת אותה** על ידי הורדת כפילויות רצופות של אותיות במחרוזת. כלומר, כל אות שמופיעה מספר פעמים ברצף במחרוזת, תצטמצם לאחר הורדת הכפילויות למופע יחיד של אות זו. דוגמאות:

המחרוזת "helllloooootoyoooouuu" תתומצת ל-"helotoyou"  
המחרוזת "strstr" תשאר ללא שינוי.

דרישות סיבוכיות: סיבוכיות זמן  $O(n)$ , סיבוכיות מקום נוסף  $O(1)$ , כאשר  $n$  אורך המחרוזת.  
**שימו לב שהפונקציה צריכה לשנות את המחרוזת המקורית ולא ליצור מחרוזת חדשה.**

```
void summarize(char *s) {
```

```
    char *p, *q;
```

```
    p = q = s;
```

```
    while (*p) {
```

```
        *q = *p;
```

```
        while (*p == *q) p++;
```

```
        q++;
```

```
    }
```

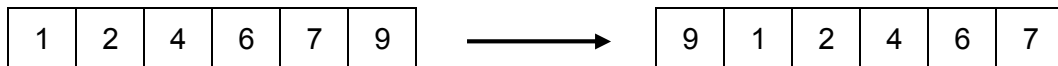
```
    *q = 0;
```

```
}
```



#### שאלה 4 (20 נקודות)

בהנתן מערך ממויין, **הסטה מעגלית** שלו תוגדר כהזזת כל איבר במערך תא אחד קדימה, כאשר האיבר האחרון במערך מועבר לתא הראשון. לדוגמה:



מערך ממויין שהופעלו עליו מספר שלם כלשהו של הסטות מעגליות (0 הסטות או יותר) ייקרא **מערך ממויין בהסטה**. לדוגמה, המערכים הבאים ממויינים בהסטה:

(הופעלו 2 הסטות) 

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

(הופעלו 4 הסטות) 

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

(הופעלו 0 הסטות) 

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

עליכם לממש את הפונקציה `findcirc()` שמקבלת מערך ממויין בהסטה (שימו לב שגודל ההסטה אינו ידוע), את גודלו `n` וערך `x`, ומחפשת בו את הערך `x`. הפונקציה מחזירה את האינדקס במערך שבו `x` נמצא, או (-1) אם `x` אינו נמצא במערך. ניתן להניח כי כל אברי המערך שונים זה מזה. **שימו לב** כי קיים פתרון הפועל בסיבוכיות זמן  $O(\log n)$ ; פתרון בסיבוכיות זמן גבוהה מזו יזכה לכל היותר בניקוד חלקי.

```
int findcirc(int a[], int n, int x) {
```

```
    int i1, i2, k=breakpoint(a,n);
```

```
    i1 = binsearch(a,k+1,x);
```

```
    i2 = binsearch(a+k+1,n-k-1,x);
```

```
    if (i1!=-1) return i1;
```

```
    if (i2!=-1) return i2+k+1;
```

```
    return -1;
```

```
}
```



```
int breakpoint(int a[], int n) {  
  
    int low=0, high=n-1, mid;  
    while (low < high) {  
        mid = (low+high)/2;  
        if (a[mid] > a[mid+1]) return mid;  
        else if ((a[mid] < a[low]) && (a[mid] < a[high])) high = mid;  
        else if ((a[mid] > a[low]) && (a[mid] > a[high])) low = mid;  
        else return high;  
    }  
    return low;  
}
```

```
int binsearch(int a[], int n, int x) {  
  
    if (n <= 0) return -1;  
    if (a[n/2] == x) return n/2;  
    if (a[n/2] > x)  
        return binsearch(a, n/2, x);  
    else {  
        int pos = binsearch(a+n/2+1, n-n/2-1, x);  
        return (pos==-1) ? -1 : pos+n/2+1;  
    }  
}
```



- אפשרות נוספת לפתרון השאלה:

```
int findcirc(int a[], int n, int x) {  
  
    int low=0, high=n-1, mid;  
  
    while (high >= low) {  
        mid = (low+high)/2;  
        if (a[mid] == x) return mid;  
  
        if (a[high] > a[low]) {  
            if (a[mid]>x) high=mid-1;  
            else low=mid+1;  
        }  
        else if (a[mid] > a[high]) {  
            if (x>=a[low] && x<=a[mid]) high=mid-1;  
            else low=mid+1;  
        }  
        else {  
            if (x>=a[mid] && x<=a[high]) low=mid+1;  
            else high=mid-1;  
        }  
    }  
  
    return -1;  
}
```





## שאלה 5 (20 נקודות)

משחק סודוקו מורכב מלוח בגודל  $9 \times 9$  של ספרות בטווח 1-9, ממולא חלקית. לדוגמה:

			1			7	4	
	5			9			3	2
		6	7			9		
4			8					
	2						1	
					9			5
		4			7	3		
7	3			2			6	
	6	5			4			

מטרת השחקן למלא את יתרת המשבצות בלוח בספרות 1-9 כך ש:

- באותה שורה לא תופיע אף ספרה יותר מפעם אחת.
- באותו טור לא תופיע אף ספרה יותר מפעם אחת.

שימו לב שבמשחק המלא יש תנאי נוסף, אך אנו לא מתייחסים אליו בשאלה זו.

לדוגמה, הנה פתרון ללוח למעלה:

3	9	2	1	8	5	7	4	6
8	5	7	4	9	6	1	3	2
1	4	6	7	3	2	9	5	8
4	7	9	8	5	1	6	2	3
5	2	8	6	7	3	4	1	9
6	1	3	2	4	9	8	7	5
2	8	4	5	6	7	3	9	1
7	3	1	9	2	8	5	6	4
9	6	5	3	1	4	2	8	7

בשאלה זו נייצג לוח סודוקו באמצעות מערך דו-ממדי `board[9][9]`, המכיל מספרים בטווח 1-9 במקומות המלאים, ואפסים במקומות הריקים. עליכם לממש את הפונקציה `() sudoku` (בדף הבא) המקבלת לוח סודוקו מלא חלקית, וממלאת אותו לפתרון מלא כלשהו על ידי החלפת האפסים שבמערך במספרים בין 1 ל-9. ניתן להניח כי הלוח שהפונקציה `() sudoku` מקבלת הינו פתיר.

**שימו לב:** על הפתרון להיות רקורסיבי ולעבוד בשיטת ה-`backtracking`; בפרט, אין להמשיך ברקורסיה עם לוח שברור כי אינו פתיר. זכרו שביכולתכם לממש פונקציות עזר נוספות על פי הצורך.



```
void sudoku(int board[9][9]) {
    sudoku_aux(board);
}

int sudoku_aux(int board[9][9])
{
    int i,j,num;
    for (i=0; i<9; ++i) {
        for (j=0; j<9; ++j) {
            if (board[i][j]==0) {
                for (num=1; num <= 9; ++num) {
                    if (legal(board,i,j,num)) {
                        board[i][j] = num;
                        if (sudoku_aux(board))
                            return 1;
                        board[i][j] = 0;
                    }
                }
                return 0;
            }
        }
    }
    return 1;
}

int legal(int board[9][9], int r, int c, int num)
{
    int k;
    for (k=0; k<9; ++k) {
        if (board[k][c] == num) return 0;
        if (board[r][k] == num) return 0;
    }
    return 1;
}
```