



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

סמסטר אביב תשס"ו

פתרון מבחן מסכם מועד א', 14 יולי 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. 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.
- ניתן להניח שכל משתנה מטיפוס מחרוזת (`char*`) מסתיימת ב-`'\0'`.
- 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 אוהיון (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 -



שאלה 1 (20 נקודות)

בדומה לסדרת פיבונצ'י, סדרת אבו-נצ'י הינה סדרת מספרים בה כל איבר הינו סכום שני האיברים שלפניו. ההבדל הוא ששני האיברים הראשונים בסדרה אינם בהכרח 1 אולם הם בהכרח חד-ספרתיים. למשל, הסדרה הבאה היא סדרת אבו-נצ'י חוקית: $1, 9, 10, 19, 29, 48, \dots$. ניתן להפוך כל מספר למחרוזת ולשרשר את אוסף המחרוזות לקבלת מחרוזת אחת. לדוגמא, עבור הסדרה הנ"ל נקבל את המחרוזת "1910192948". יש לכתוב פונקציה המקבלת מחרוזת המורכבת מספרות ובודקת האם היא מכילה מספרים המקיימים את תנאי סדרת אבו-נצ'י. למשל על הפונקציה להשיב אמת בהינתן המחרוזת "1910192948" ושקר בהינתן המחרוזת "15712195069".

הערות:

- ניתן להניח כי המחרוזת מכילה ספרות בלבד.
- מחרוזת בת שתי ספרות בלבד מקיימת את התנאי (באופן ריק).
- ניתן להניח כי במחרוזת יש לפחות שתי ספרות.
- ניתן להניח כי אורך המחרוזת המקסימלית הוא N המוגדר ב `#define`.
- אין להשתמש בפונקציות ספריה.

א. ממשו את הפונקציה `num2str`, אשר מקבלת מספר שלם אי-שלילי `num` ומחזירה מחרוזת `str` המייצגת את אותו מספר בפורמט עשרוני (ניתן להניח כי הוקצה מספיק מקום עבור אותיות המחרוזת).

```
void num2str(unsigned int num, char *str) {
```

```
    int dignum = 1, i;
    unsigned int tempnum = num;

    while (tempnum > 9) {
        tempnum /= 10;
        dignum++;
    }

    str[dignum] = 0;
    while (dignum > 0) {
        dignum--;
        str[dignum] = num % 10 + '0';
        num /= 10;
    }
}
```



- 4 -



ב. ממשו את הפונקציה `startswith` אשר משווה בין מחרוזת `A` למחרוזת `B`. הפונקציה תחזיר 1 אם המחרוזת `B` מתחילה במחרוזת `A`, ו-0 אחרת. לדוגמא: עבור `A="abc"` ו `B="abcdef"` הפונקציה תחזיר 1, ועבור `B="acbdef"` הפונקציה תחזיר 0. אם `A` ריקה, הפונקציה תחזיר 1.

```
unsigned int startswith(char *A, char *B) {
```

```
    while (*A && *B) {  
        if (*A != *B)  
            break;  
        A++; B++;  
    }
```

```
    if (*A == 0)  
        return 1;
```

```
    return 0;
```

```
}
```

ג. ממשו את הפונקציה `IsAbuNacci`, אשר מחזירה 1 אם מחרוזת הקלט `str` מקיימת את תנאי סדרת אבו-נצ'י, ו-0 אחרת. מותר (ואף מומלץ) להשתמש בפונקציות שהוגדרו בסעיפים א' ו-ב', גם אם לא פתרתם אותם.

```
unsigned int IsAbuNacci(char *str) {
```

```
    int num1, num2, pos=0;  
    char nextnum[N];
```

```
    num1 = str[pos++] - '0';  
    num2 = str[pos++] - '0';
```

```
    while(str[pos]) {  
        num2str(num1+num2, nextnum);  
        if (!startswith(nextnum, str+pos))  
            return 0;  
        pos += strlen(nextnum);  
        num2 = num1+num2;  
        num1 = num2-num1;
```

```
    }
```

```
    return 1;
```

```
}
```



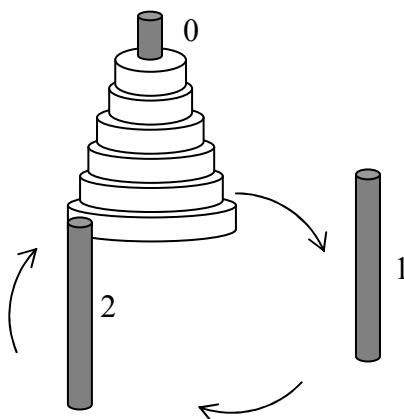
```
int strlen(char *str)
{
    int len = 0;
    while (str[len])
        len++;
    return len;
}
```



שאלה 2 (20 נקודות)

שאלה זו היא וריאציה על בעיית מגדלי הנוי.

נתונים שלושה מוטות מסודרים על מעגל. על אחד מהם מושחלות hn טבעות בגדלים שונים בערמה, כך שמעל כל טבעת (מלבד העליונה ביותר) שוכבת טבעת קטנה ממנה. כמו בבעיית מגדלי הנוי המקורית, בכל שלב לא תונח טבעת על טבעת קטנה ממנה. בנוסף לכך, העברת הטבעות נעשית רק בין מוטות סמוכים **עם כיוון השעון בלבד**. כלומר, מותר להעביר טבעת רק ממגדל 0 למגדל 1, ממגדל 1 למגדל 2, וממגדל 2 למגדל 0. אין להעביר טבעת ישירות ממגדל 0 למגדל 2.



כתבו פונקציה **רקורסיבית** בשם `CyclicHanoi` שמקבלת כפרמטר את מספר הטבעות n , את מגדל המקור `source`, ומגדל היעד `target` כמספרים שלמים (0, 1, או 2), ומדפיסה את הפתרון לבעיה. ניתן להניח כי `source \neq target`. ניתן להשתמש בפונקציה `move` אשר מדפיסה את ההודעה המתאימה להעברת טבעת ממגדל `source` למגדל `target` אך ורק אם ההעברה היא בכיוון השעון. חתימת הפונקציה `move` מוגדרת כ:

```
void move(unsigned int source, unsigned int target);
```

אנא ממשו את הפונקציה `CyclicHanoi` בדף הבא.



- 8 -



```
void CyclicHanoi(unsigned int n, unsigned int source, unsigned int target) {
```

```
    unsigned int aux = 3-source-target;
```

```
    if (n == 0)
```

```
        return;
```

```
    if (next(source) == target)
```

```
    {
```

```
        CyclicHanoi(n-1, source, aux);
```

```
        move(source, target);
```

```
        CyclicHanoi(n-1, aux, target);
```

```
    }
```

```
    else {
```

```
        CyclicHanoi(n-1, source, target);
```

```
        move(source, aux);
```

```
        CyclicHanoi(n-1, target, source);
```

```
        move(aux, target);
```

```
        CyclicHanoi(n-1, source, target);
```

```
    }
```

```
}
```

```
int next(int polenum){
```

```
    return (polenum+1)%3;
```

```
}
```



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



שאלה 3 (20 נקודות)

סטודנט בקורס למדעי המחשב החליט לפתח אלגוריתם מיון חדש. שיטת המיון שלו עובדת בצורה הבאה: בהנתן מערך של מספרים A , נייצג אותו בצורה של מטריצה עם m עמודות, כאשר ממלאים את השורות משמאל לימין עם איברי A לפי הסדר. נמייין כל עמודה במטריצה בנפרד ונחזיר לייצוג של מערך. נחזור על התהליך עם ערכי m הולכים וקטנים, עד אשר המערך יהיה ממויין.

לדוגמא, עבור המערך: $A = \{7, 9, 1, 2, 8, 8, 9, 4, 3, 7, 2\}$

ו- $m = 3$, נקבל את המטריצה הבאה (צד שמאל), ולאחר מיון עמודות (צד ימין)

7	9	1
2	8	8
9	4	3
7	2	

→

2	2	1
7	4	3
7	8	8
9	9	

או, אם נחזיר לייצוג של מערך: $\{2, 2, 1, 7, 4, 3, 7, 8, 8, 9, 9\}$

כעת, נייצג את המערך שהתקבל כמטריצה עם 2 עמודות, ונמייין כל עמודה:

2	2
1	7
4	3
7	8
8	9
9	

→

1	2
2	3
4	7
7	8
8	9
9	

או, אם נחזיר לייצוג של מערך: $\{1, 2, 2, 3, 4, 7, 7, 8, 8, 9\}$ (ממויין)

המוטיבציה לאלגוריתם זה היא התקווה כי בכל שלב של מיון עמודות, מספר האיברים שנמצאים במקומם הנכון הולך וגדל (במילים אחרות, אי-הסדר קטן).

- א. ממשו את הפונקציה `SortByMatrixColumn` אשר מקבלת מערך מספרים A בגודל n ומספר טבעי m המציין את מספר העמודות (ביצוג המטריצה), וממבצעת שלב אחד של אלגוריתם המיון (עבור ערך m נתון). על הפונקציה לרוץ בסיבוכיות זמן $O(n^2/m)$ ובסיבוכיות מקום $O(1)$.
רמז: ניתן לממש פונקציית עזר המקבלת קואורדינטות במטריצה עם m עמודות ומחזירה את האינדקס המתאים במערך.

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

```
void SortByMatrixColumn (int *A, unsigned int n, unsigned int m) {
```

```
int i, j, col, minid;

for (col=0; col<m; col++)
{
    for (i=col; i<n; i+=m)
    {
        minid = i;
        for (j=i+m; j<n; j+=m)
            if (A[j] < A[minid])
                minid = j;
        swap(A+i, A+minid);
    }
}
```

```
void swap(int *x, int *y)
{
    int tmp = *x;
    *x = *y;
    *y = tmp;
}
```

This image shows a blank 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.



- 14 -



שאלה 4 (20 נקודות)

כתבו פונקציה `translate` המקבלת מחרוזת `str`, ושתי מחרוזות נוספות `orig` ו-`new`. כל המחרוזות מכילות אותיות בתחום 'a'-'z' בלבד. הפונקציה משנה את המחרוזת המקורית `str`, כך שבכל מקום בו הופיע תו מ-`orig`, הוא יוחלף בתו המקביל מ-`new`. תווים ב-`str` שלא מופיעים ב-`orig` לא יושפעו. אם תו מופיע ב-`orig` יותר מפעם אחת, המופע האחרון קובע את כלל ההחלפה.

דוגמאות

מחרוזת חדשה	str	המרות	new	orig
xyookonz	abookonc	'a' → 'x' 'b' → 'y' 'c' → 'z'	xyz	abc
dyookonz	abookonc	'a' → 'd' 'b' → 'y' 'c' → 'z'	xyzd	abca

על הפונקציה לרוץ בסיבוכיות זמן $O(n+m)$, כאשר n הוא אורך המחרוזת `str`, ו- m הוא אורך המחרוזת `orig` ו-`new`, ובסיבוכיות מקום $O(1)$.

הנחיות נוספות:

- ניתן להניח שהמחרוזת `orig` ו-`new` בעלות אותו אורך.
- המחרוזות אינן ממויינות בשום צורה שהיא.
- אין להשתמש בפונקציות ספרייה.

```
void translate(char *str, char *orig, char *new) {

    int convert_table['z'-'a'+1] = {0};

    while (*orig) {
        convert_table[*orig-'a'] = *new;
        orig++;
        new++;
    }

    while (*str) {
        if (convert_table[*str-'a']) {
            *str = convert_table[*str-'a'];
        }
        str++;
    }
}
```



- 16 -



שאלה 5 (20 נקודות)

הסיעות השונות במועצה מנסות להרכיב קואליציה חדשה. קואליציה תקפה הינה קבוצה של סיעות, אשר כל אחת מהן מוכנה לשתף פעולה עם כל יתר הסיעות בקבוצה, וכן, סך חברי כל הסיעות בקבוצה עולה על מחצית סך חברי המועצה.

במועצת הארגון יש N סיעות, ו- M חברים סך-הכל $(N \text{ ו- } M)$ הם קבועים המוגדרים באמצעות `#define`. נתון מערך `parties` המכיל את מספר חברי המועצה מכל סיעה – מספר חברי הסיעה ה- i במועצה מאוכסן בתא `parties[i]`.

כמו-כן נתונה מטריצה בשם `relation`, המתארת את היחסים בין הקבוצות השונות: התא `relation[i][j]` מכיל 1 אם ורק אם הסיעה ה- i מוכנה לשתף פעולה עם הסיעה ה- j , ו-0 אחרת. שימו לב כי המטריצה r אינה בהכרח סימטרית (ייתכן שסיעה i מוכנה לשתף-פעולה עם סיעה j , אך j אינה מוכנה לשתף פעולה עם i). ניתן להתעלם מן התאים באלכסון של המערך (כי כל סיעה מוכנה להשתתף בקואליציה עם עצמה).

עליכם לכתוב פונקציה בשם `findc`, אשר מחזירה את מספר החברים בקואליציה התקפה הגדולה ביותר האפשרית במועצה. אם לא קיימת אף קואליציה תקפה, על הפונקציה להחזיר -1.

יש לפתור את הבעיה בשיטת BackTracking. פתרון רקורסיבי ללא BackTracking יקבל ניקוד חלקי בלבד.

דוגמא: עבור הקבועים הבאים:

```
#define N 4
#define M 120
```

ומטריצה `relation` הבאה:

1	1	1	0
1	1	1	0
1	1	1	0
1	0	0	1

ומערך `parties` הבא:

50
20
20
30

על הפונקציה להחזיר 90 (כי מפלגות 0,1,2 מוכנות להקים קואליציה, וסה"כ המנדטים שלהם הוא 90).

[illegible]



```
int findc(unsigned int parties[N], unsigned int relation [N][N]) {

    int chosen[N] = {0}, size;
    size = findc_aux(parties, relation, chosen, 0);
    return (size > M/2) ? size : -1;
}

int findc_aux(int parties[N], int relation[N][N],
              int chosen[N], int id)
{
    int i, sum, size1=0, size2=0;

    if (id==N) {
        sum = 0;
        for (i=0; i<N; ++i)
            sum += parties[i]*chosen[i];
        return sum;
    }

    if (islegal(chosen, relation, id)) {
        chosen[id] = 1;
        size1 = findc_aux(parties, relation, chosen, id+1);
        chosen[id] = 0;
    }
    size2 = findc_aux(parties, relation, chosen, id+1);
    return (size1 > size2) ? size1 : size2;
}

int islegal(int chosen[N], int relation[N][N], int id)
{
    int i;
    for (i=0; i<id; ++i) {
        if (chosen[i] && (relation[i][id]==0 ||
                        relation[id][i]==0))
            return 0;
    }
    return 1;
}
```



- 20 -



- 21 -



- 22 -