



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

סמסטר חורף תשע"ב

מבחן מסכם מועד א', 9 פברואר 2012

<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.

הנחיות והוראות:

- 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 ומחק, 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 אלא אם צויין אחרת בשאלה, למעט פונקציות קלט/פלט והקצאת זיכרון דינמי.

צוות הקורס 234114/7

מרצים: דן רביב, רן רובינשטיין, רון קימל (מרצה אחראי).

מתרגלים: אלכסנדר נוס, אריאלה וולושין, שי גרץ, אנסטסיה דוברובינה, עמרי אייזנקוט, דן גרבר, חסן עבסי, שחר תמנת, חביאר טורק (מתרגל אחראי).

בהצלחה!



- 2 -



שאלה 1 (25 נקודות)

בשאלה זו נממש פונקציה שמקבלת מחרוזת קלט s המכילה אותיות אנגליות קטנות וגדולות בלבד, ומקודדת אותה באמצעות מספר סודי (שלם וחיובי) x , כך שרק מי שידע את המספר הסודי יכול לפענח את המחרוזת.

לצורך הקידוד נשתמש בהצפנה פשוטה הנקראית צופן קיסר. בהצפנה זו, כל אות במחרוזת s מוחלפת באות שנמצאת x מקומות אחריה בא"ב האנגלי. למשל, אם $x=2$, האות a מוחלפת ל- c , והאות d מוחלפת ל- f . במידה ונוצרת חריגה מעבר לתחום האותיות האנגליות, הקידוד הוא סיבובי – כלומר, אחרי האות z באה האות a , וכן הלאה. לפיכך, עבור הדוגמה $x=2$ למשל, האות y מוחלפת ל- a , והאות z מוחלפת ל- b .

הערות:

- אותיות קטנות צריכות להישאר קטנות, ואותיות גדולות צריכות להישאר גדולות.
- ניתן להניח שמחרוזת הקלט חוקית – מכילה אותיות אנגליות בלבד.
- על הפונקציה לשנות את מחרוזת הקלט עצמה.
- יש לפתור את השאלה בסיבוכיות זמן $O(n)$ וסיבוכיות מקום $O(1)$, כאשר n הוא אורך המחרוזת.

דוגמה: אם s מכילה את המחרוזת "YesWeCan" אזי הקריאה `cipher(s,3)` תשנה אותה ל-"BhvZhFdq".

```
void cipher(char* s, unsigned int x) {  
    int ABClen = 'z' - 'a' + 1;  
  
    while (*s) {  
        if ((*s >= 'a') && (*s <= 'z')) {  
            *s = 'a' + (*s - 'a' + x) % ABClen;  
        }  
        else if ((*s >= 'A') && (*s <= 'Z'))  
        {  
            *s = 'A' + (*s - 'A' + x) % ABClen;  
        }  
        s++;  
    }  
}
```



- 4 -



שאלה 2 (25 נקודות)

סעיף א

נתונה הפונקציה הבאה:

```
void interesting(int n)
{
    int* b = NULL;
    int m, i;

    m = 0;
    i = n;
    while (i > 1)
    {
        m += n;
        i /= 2;
    }

    printf("%d\n", m);

    i = n;
    while (i > 4)
    {
        b = (int*)malloc(n*sizeof(int));
        for (j=0; j<m; ++j)
        {
            b[j%n] = j;
        }
        i /= 5;
        free(b);
    }
}
```

1. מה תדפיס שורת ה-`printf` בפונקציה עבור הקריאה `interesting(16)` ? 64

2. השלימו את סיבוכיות הזמן והמקום של הפונקציה כתלות ב-`n`:

סיבוכיות זמן: $\Theta(n \log^2(n))$ סיבוכיות מקום נוסף: $\Theta(n)$

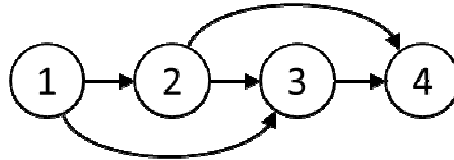


- 6 -



סעיף ב

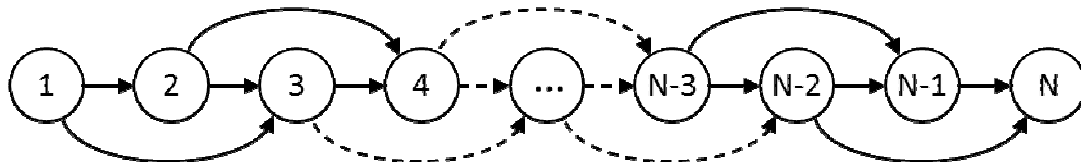
1. נתון הגרף הבא, המורכב מארבעה קודקודים וחמש קשתות:



נשים לב שבגרף זה אפשר להגיע מקודקוד אחד לאחר במספר מסלולים. למשל, מקודקוד 1 לקודקוד 3 אפשר להגיע **בשני מסלולים שונים**: או במסלול $3 \leftarrow 2 \leftarrow 1$, או במסלול $3 \leftarrow 1$.

בכמה מסלולים שונים זה מזה אפשר להגיע מקודקוד 1 לקודקוד 4 ? 3

2. כעת נתבונן בגרף הבא, הבנוי באופן דומה, אך מורכב מ-N קודקודים. שימו לב שמכל קודקוד בגרף יוצאות שתי קשתות לשני הקודקודים שאחריו:

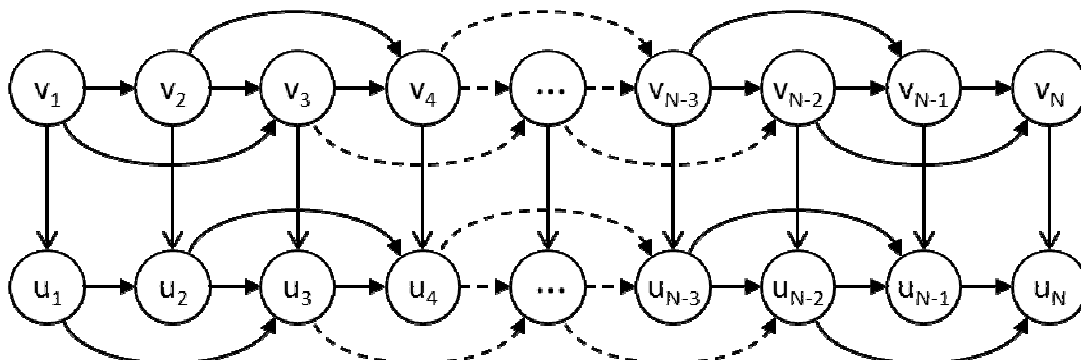


נסמן את מספר המסלולים השונים בגרף המובילים מקודקוד 1 לקודקוד N ב-N- $f(N)$. כתבו נוסחה רקורסיבית לחישוב $f(N)$ (תוך שימוש ב- $f(k)$ עבור $k < N$):

הערות: א. שימו לב שאין צורך לפתח את הנוסחה ולהגיע לפתרון סופי – רק לכתוב את נוסחת הנסיגה. ב. יש לכתוב ביטוי פשוט לנוסחת הנסיגה, שאינו כולל סכום סדרה (דהיינו, אין להשתמש ב- Σ_i).

$$f(N) = f(N-1) + f(N-2)$$

3. לבסוף, נתון הגרף הבא, הדומה לקודם, אך בו יש גם קשת מכל קודקוד בשורה העליונה לקודקוד המתאים בשורה התחתונה:



נסמן ב- $g(N)$ את מספר המסלולים השונים בגרף המובילים מקודקוד v_1 (שמאל למעלה) לקודקוד u_N (ימין למטה). כתבו נוסחה רקורסיבית לחישוב $g(N)$. שימו לב שהערות א' ו-ב' למעלה תקפות גם כאן. בנוסף, הנוסחה יכולה להשתמש גם בפונקציה f במידת הצורך:

$$g(N) = g(N-1) + g(N-2) + f(N)$$



- 8 -



שאלה 3 (25 נקודות)

בשאלה זו נתון מערך $a[]$ באורך n (לא בהכרח ממין) המכיל מספרים שלמים. כמו כן ידוע כי ערכי המערך כולם בטווח בין 0 ל- $k-1$ (כולל).

נאמר שהמערך $a[]$ חוקי אם אין בו ערכים שחוזרים על עצמם, וכן אין בו ערכים עוקבים. כלומר, אם הערך x מופיע במערך, אזי הוא מופיע בו בדיוק פעם אחת, והערכים $x-1$, $x+1$ לא מופיעים כלל במערך.

למשל, המערך הבא הוא חוקי: $a[] = \{ 5, 1, 8, 17 \}$

לעומת זאת, המערכים הבאים אינם חוקיים:

$b[] = \{ 5, 1, 8, 5 \}$ (הערך 5 חוזר על עצמו)

$c[] = \{ 5, 1, 8, 2 \}$ (1 ו-2 מספרים עוקבים)

ממשו את הפונקציה בעמוד הבא, שמקבלת את המערך $a[]$, את אורכו n , ואת הערך k , ומחזירה 1 במידה והמערך חוקי, או 0 אחרת. שימו לב שמערך ריק הוא חוקי.

הערות:

- על הפונקציה לעמוד בסיבוכיות זמן $O(n+k)$. אין הגבלה על סיבוכיות המקום.
- פתרונות בסיבוכיות זמן גרועה מהנדרש יזכו לניקוד חלקי במידה והם נכונים.
- שימו לב שבשאלה זו k אינו נחשב כקבוע לצרכי חישובי סיבוכיות.
- בנוסף לפתרון, ענו בקצרה על שתי השאלות הבאות:

א. הסבירו בקצרה מדוע הפתרון שלכם עומד בדרישת סיבוכיות הזמן של $O(n+k)$. לחילופין, במידה ופתרתם בסיבוכיות זמן גבוהה מהנדרש, כתבו את סיבוכיות הזמן של הפתרון שלכם, והסבירו כיצד חיבתם אותה:

תשובה:

סיבוכיות הזמן – $\Theta(n+k)$.

ישנו מעבר אחד על המערך a שגודלו n (לשם בניית hist) ושני מעברים על המערך hist שגודלו k (האחד לאתחולו והשני לבדיקה של תנאי החוקיות). בכל איטרציה מתבצע מספר קבוע של פעולות. בקירוב אסימפטוטי תוך הזנחת הקבועים ושאר הפעולות שאינן תלויות ב- n וב- k מתקבלת סיבוכיות של $\Theta(n+k)$.

ב. מהי סיבוכיות המקום הנוסף של הפתרון שכתבתם?

סיבוכיות מקום נוסף: $\Theta(\text{ } k \text{ })$



- 10 -



```
int islegal(int a[], unsigned int n, unsigned int k) {
    unsigned int i = 0;
    int *hist;
    if (n <= 1)
        return 1;
    hist = (int*)malloc(sizeof(int)*k);
    if (hist == NULL) {
        printf("Allocation error\n");
        return -1;
    }
    for (i=0; i<k; i++)
        hist[i] = 0;

    for (i=0; i<n; i++)
        hist[a[i]]++;

    for (i=0; i<k; i++)
        if ((hist[i]>1) ||
            (i<k-1 && hist[i]==1 && hist[i+1]>0)) {
            free(hist);
            return 0;
        }

    free(hist);
    return 1;
}
```



- 12 -



שאלה 4 (25 נקודות)

באי הנידח Sfatsea שבאוקיאנוס ההודי חיים N אנשים, המקיימים ביניהם מערכת חברתית סבוכה: כל תושב באי משוייך למעמד מסוים, ובשל פערי מעמדות, מותר לו להתגורר רק עם תושבים אחרים בני אותו המעמד. לאחרונה החליטו באי לעבור להתגורר בבתים חדשים, ולצורך כך הם בנו k בתים. כעת, עליהם להחליט באיזה בית יגור כל תושב באי, תוך התחשבות במעמדות.

בשאלה זו נכתוב פונקציה המסייעת לתושבי האי למצוא סידור מגורים. הקלט של הפונקציה הינו מספר הבתים שנבנו k , מספר האנשים המקסימאלי בכל בית m , וכן מטריצה $a[N][N]$ (מוגדר כ-`#define`) המתארת למי מותר לגור עם מי: $a[i][j]$ מכיל 1 במידה והתושבים i ו- j יכולים לגור יחד, ו-0 אם לא. שימו לב שהמטריצה a סימטרית, כלומר אם $a[i][j]==1$ אז בהכרח גם $a[j][i]==1$.

על הפונקציה למצוא חלוקת חוקית כלשהי של התושבים לבתים, או להודיע שלא קיימת כזו. במידה וישנה חלוקה חוקית, הפונקציה מחזירה 1, וכותבת חלוקה אפשרית כלשהי לתוך מערך הפלט $h[]$ שאורכו N , כאשר $h[i]$ מכיל את מספר הבית שאליו משוייך התושב i , בין 0 ל- $k-1$. במידה ואין כל חלוקה חוקית, הפונקציה מחזירה 0 ואין חשיבות לתוכן המערך $h[]$.

הערות:

- על הפתרון להיות רקורסיבי ולהשתמש ב-backtracking.
- אין להשתמש במשתנים סטטיים.
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-backtracking, יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.
- ניתן בהחלט להגדיר ולממש פונקציות עזר על מנת לפשט את הקוד.

```
int housing(int a[N][N], int k, int m, int h[]) {
    return housing_aux(a,k,m,h,0);
}

int housing_aux(int a[N][N], int k, int m, int h[], int p)
{
    int house, i, count;
    if (N == p) return 1;
    for (house=0; house<k; house++) {
        count = 0;
        for (i=0; i<p; i++) {
            if (h[i] == house) {
                if (!a[p][i]) {
                    count = m;
                    break;
                }
            }
            count++;
        }
    }
}
```



```
    }  
    if (count >= m) continue;  
    h[p] = house;  
    if (housing_aux(a,k,m,h,p+1))  
        return 1;  
}  
return 0;  
}
```



- 15 -



- 16 -



- 17 -



- 18 -