



## מבוא למדעי מחשב ח' / (234117)

### סמסטר אביב תשע"ה

### מבחן מסכם מועד א', 6 יולי 2015

2	3	4	1	1	
---	---	---	---	---	--

רשום/ה לקורס:

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

מספר סטודנט:

משך המבחן: 3 שעות.

חומר עזר: אין להשתמש בכל חומר עזר.

#### הנחיות כלליות:

- מלאו את הפרטים בראש דף זה ובדף השער המצורף, בעט בלבד.
- בדקו שיש 15 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 לכתיבת 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.
- אלא אם כן נאמר אחרת בשאלות, **אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה**, למעט פונקציות קלט/פלט והקצאת זיכרון (`malloc`, `free`). ניתן להשתמש בטיפוס `bool` המוגדר ב-`stdbool.h`.
- אין להשתמש במשתנים סטטיים וגלובאליים אלא אם נדרשתם לכך מפורשות.
- ניתן להשתמש בהקצאות זכרון בסגנון C99 (מערכים בגודל משתנה), בכפוף לדרישות סיבוכיות זכרון.
- כשאתם נדרשים לכתוב קוד באילוצי סיבוכיות זמן/מקום נתונים, אם לא תעמדו באילוצים אלה תוכלו לקבל בחזרה מקצת הנקודות אם תחשבו נכון ותציינו את הסיבוכיות שהצלחתם להשיג.
- נוסחאות שימושיות:

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \Theta(\log n) \quad 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots = \Theta(1)$$

$$1 + 2 + \dots + n = \Theta(n^2) \quad 1 + 4 + 9 + \dots + n^2 = \Theta(n^3) \quad 1 + 8 + 27 + \dots + n^3 = \Theta(n^4)$$

צוות הקורס 234114/7

**מרצים:** פרופ' ניר אילון (מרצה אחראי), פרופ' מירי בן-חן, ד"ר רוני קופרשטוק **מתרגלים:** מר נחשון כהן (מתרגל ראשי), גב' דניאל עזוז, גב' גילי יבנה, מר שרגא לבציון, מר מאור ינקוביץ', מר יפתח זיסר, מר אחאב וואטאד

**בהצלחה!**



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

[illegible]



### שאלה 1 (25 נקודות):

א. (8 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה  $f$  המוגדרת בקטע הקוד הבא, כפונקציה של  $n$ . אין צורך לפרט שיקולים. חובה לפשט את הביטוי ככל שניתן.

```
void f(int n)
{
    int s=1;
    for(int i=0; i<n; ++i)
        s*=2;
    for(int i=0; i<s; ++i)
        for(int j=0; j<i; ++i)
            printf("~");
}
```

סיבוכיות זמן:  $\Theta(\quad)$       סיבוכיות מקום:  $\Theta(\quad)$

ב. (9 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה  $g()$

```
int g(int n){
    int q;
    for(q=0; q*q<n; ++q)
        ;
    int vals[q];
    for(int i=0; i<n*n; i+=q){
        vals[i/(n*n)]+=q;
        printf("!");
    }
    return vals[0];
}
```

סיבוכיות זמן:  $\Theta(\quad)$       סיבוכיות מקום:  $\Theta(\quad)$

ג. (8 נקודות) חשבו את סיבוכיות הזמן והמקום של הפונקציה  $w()$  (ולא של הפונקציה  $w\_aux$ )

```
void w_aux(int n){
    if(n<2)
        return;
    for(int i=0; i<n; ++i)
        printf("#");
    w_aux(n/2);
    w_aux(n/2);
}
void w(int n){
    w_aux(n*n*n*n);
}
```

סיבוכיות זמן:  $\Theta(\quad)$       סיבוכיות מקום:  $\Theta(\quad)$

[illegible]



## שאלה 2 (25 נקודות) :

ממשו את הפונקציה הבאה, שחתימתה

```
void sortPartialSorted(int a[], int n, int k);
```

הפונקציה מקבלת כקלט מערך  $a$  של מספרים באורך  $n$  ועוד מספר שלם  $k \leq \sqrt{n}$ . ידוע שכל המספרים ממיקום  $k$  ואילך ממויינים, אבל לא ידוע דבר על המספרים במיקום  $0..k-1$ . הפונקציה צריכה למיין את המערך. לדוגמא, אם  $a$  הוא המערך הבא:

```
int a[9] = {17, 1, 23, 1, 4, 6, 9, 12, 20};
```

וכן  $n=9$ ,  $k=3$ , אז הפונקציה צריכה להחזיר את המערך הבא:

1	1	4	6	9	12	17	20	23
---	---	---	---	---	----	----	----	----

**דרישות:** סיבוכיות זמן  $O(n)$  וסיבוכיות מקום נוסף  $O(\sqrt{n})$ .

אם לפי חישוביכם לא עמדתם בדרישות הסיבוכיות, אנא ציינו כאן את הסיבוכיות שהגעתם אליה: זמן \_\_\_\_\_ מקום נוסף \_\_\_\_\_

```
void sortPartialSorted(int a[], int n, int k)
```

```
{
```

[illegible]



### שאלה 3 (25 נקודות):

עליכם לממש את הפונקציה:

```
void trimSpecial(char *str, char *special);
```

הפונקציה מקבלת כקלט מחרוזת `str` ועוד מחרוזת של תווים מיוחדים. הפונקציה צריכה למחוק מהמחרוזת `str` את כל התווים המיוחדים שמופיעים ב `special`.  
לדוגמה: עבור המחרוזת `str="good luck in the exam!!!"` ו `special="o!"`, לאחר הקריאה לפונקציה המחרוזת `str` תכיל `"gd luck in the exam"`.  
אם `special="aeiou"` (שימו לב ש `special` מכילה רווח) ו `str` כמו בדוגמה קודמת אז לאחר הקריאה `str="gd!cknthxm!!!"`.

**מותר** לכם להניח שכל התווים במחרוזת `special` בעלי ערך אסקיי גדול ממש מ 0 וקטן ממש מ 128.

**דרישות:** סיבוכיות זמן  $O(n+m)$  וסיבוכיות מקום נוסף  $O(1)$ , כאשר  $n$  הוא מספר התווים הכולל ב `str` ו  $m$  מספר התווים הכולל ב `special`.

אם לפי חישוביכם לא עמדתם בדרישות הסיבוכיות, אנא ציינו כאן את הסיבוכיות שהגעתם אליה: זמן \_\_\_\_\_ מקום נוסף \_\_\_\_\_

```
void trimSpecial(char *str, char *special)
```

```
{
```


[illegible]





#### שאלה 4 (25 נקודות) :

בטיסה לפורטלנד ישנם  $N$  נוסעים ו- $M$  מקומות ישיבה במטוס, כאשר  $N \leq M$  (כלומר, יתכנו מקומות ישיבה ללא נוסע). בנוסף כל נוסע נותן 2 העדפות לגבי מיקום הישיבה שלו. עליכם לכתוב פונקציה המקבלת את ההעדפות הנוסעים ומוציאה סידור ישיבה כך שכל נוסע יקבל את אחד המקומות המועדפים עליו.

הקלט הוא מערך בעל  $N$  שורות ו-2 עמודות. השורה  $i$  מכילה את 2 ההעדפות של נוסע  $i$ . העדפה מיוצגת על ידי מספר המקום (מספר בין 0 ל  $M-1$ ).

עליכם לכתוב פונקציה

```
bool findSeats(int preferences[N][2], int res[N]);
```

המקבלת את ההעדפות ומערך  $res$  וכותבת עבור כל אחד מהנוסעים את המיקום שלו. אם לא קיימת דרך להושיב את הנוסעים יש להחזיר  $false$ . אחרת יש להחזיר  $true$ .

לדוגמה: עבור ההעדפות  $N=4, M=4$

0	0	1
1	0	2
2	1	3
3	3	0

יש להחזיר  $true$ . המערך  $res$  עשוי להכיל

0	2	1	3
---	---	---	---

(נוסעים 0,2,3 קיבלו את ההעדפה הראשונה שלהם, ונוסע 1 קיבל את ההעדפה השניה).

עבור ההעדפות  $N=4, M=5$

0	0	1
1	0	1
2	0	1
3	0	4

יש להחזיר  $false$  כי לא קיימת דרך להושיב את כולם.

#### הערות:

- $N$  מוגדר ע"י `#define`.
- יש להשתמש בשיטת `backtracking` כפי שנלמדה בכיתה.
- אפשר להניח ששתי ההעדפות של כל נוסע שונות.
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-`backtracking` יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.

 $\{$

[illegible]

[illegible]

[illegible]



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

[illegible]

[illegible]