מבוא למדעי המחשב מ'/ח' (234114 \ 234117)

חורף תשפ"ד

מועד א סבב ראשון, 9.5.2024

מספר סטודנט: | | | | | | | | | |

רשום/ה לקורס: | | | | | | |
|---|---|---|---|---|---|
| | 1 | 1 | 4 | 3 | 2 |

משך המבחן: שלוש שעות.
חומר עזר: אין להשתמש בכל חומר עזר.

הנחיות כלליות:

- בדקו שיש 11 עמודים (4 שאלות) במבחן, כולל עמוד זה.
- אלא אם כן נאמר אחרת בשאלות, **אין להשתמש בפונקציות ספריה או בפונקציות שמומשו בכיתה**, למעט פונקציות קלט ופלט והקצאת זיכרון (`malloc`, `free`). ניתן להשתמש בטיפוס `bool` המוגדר ב־`stdbool.h`.
- אין להשתמש במשתנים סטטיים וגלובליים אלא אם נדרשתם לכך מפורשות.
- הקפידו על סגנון כתיבה כזה שקורא התוכנית ידע לפענח את רעיונותיכם. אי הקפדה על סגנון, עלולה להוביל להורדת נקודות.
  - א. התוכנית יכולה להכיל תיעוד קל להבנה.
  - ב. על התוכנית להיות כתובה באופן מסודר ומדורג.
  - ג. יש לתת שמות פונקציות ושמות משתנים משמעותיים.
  - ד. ערכים קבועים יש להגדיר באמצעות `define` .
  - ה. ניתן להוסיף פונקציות עזר כרצונכם.
- שימו לב שקיבלתם את קבצי ה־ C ובהם שלד של התוכנית. עליכם רק להשלים את המימוש של הפונקציה הנדרשת.
- לכל שאלה קיבלתם מספר טסטים. מומלץ להוסיף טסטים שלכם.
- נוהל "לא יודע": אם תורידו את ההערה מהקריאה לפונקציה `printIDontKnow()` על שאלה שבה אתם נדרשים לקודד, תקבלו 20% מהניקוד. דבר זה מומלץ אם אתם יודעים שאתם לא יודעים את התשובה.

הנחיות הגשה: חשוב לקרוא בעיון!

1. לפני תחילת המבחן, יש למלא במודל **הצהרה על טוהר הבחינות**. סיסמא להצהרה במודל תינתן במעמד הבחינה. ניתן למלא את ההצהרה רק בחצי השעה הראשונה.
2. הגשת כל שאלה במטלה נפרדת במערכת Gradescope לפי השלבים הבאים:
   a. ניתן להגיש מטלות רק לאחר ההצהרה על טוהר הבחינות. ולאחר חצי שעה מתחילת המבחן.
   b. אין להיכנס למטלות המבחן ב Gradescope מחוץ לכיתות הלימוד.
   c. יש להתחבר לGradescope אך ורק דרך הקישור המופיע באתר הקורס. (למעט מקרים חריגים באישור מפורש מצוות הקורס). כניסה שלא דרך המודל ולא אושרה מפורשות על ידי צוות הקורס תגרור העלאה לוועדת משמעת.
3. יש למלא **הצהרה על סיום הבחינה** תוך 5 דקות מהגשת המטלה האחרונה שפתרתם. הזמנים ייבדקו באופן אוטומטי.
4. במהלך המבחן מותר לעבוד אך ורק בסביבת CLion.
5. אין להשתמש בשום כלי בינה מלאכותית יוצרת כגון: ChatGPT, CoPilot וכדומה. שימוש בכלים אלו יביא להעלאה לוועדת משמעת.
6. הגשה לא לפי השלבים שפורטו לעיל, ובפרט: אי הצהרה על טוהר הבחינות\הצהרה על סיום הבחינה, או הגשת מטלה לאחר הצהרה על סיום הבחינה, או הצהרה מאוחרת מידי על סיום הבחינה, **תוביל לציון 0 בקורס והעלאה לוועדת משמעת**.

נוסחאות שימושיות:

$$1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}=\Theta(\log n)$$

$$1^k+2^k+3^k+\cdots+n^k=\Theta(n^{k+1})$$

$$\log 1+\log 2+\log 3+\cdots \log n=\Theta(n\cdot\log n)$$

$$1+k+k^2+k^3+k^n=\Theta(k^n), k>1$$

סכום סדרה חשבונית: $S_n=n\cdot\frac{a_1+a_n}{2}$

סכום סדרה הנדסית: $S_n=a_1\cdot\frac{q^n-1}{q-1}$

סכום סדרה הנדסית אינסופית: $S_n=\frac{a_1}{1-q}$.

בהצלחה :)

צוות הקורס 234114/7

מרצים: גב' יעל ארז(מרצה אחראית), ד"ר יוסי וינשטיין, ד"ר לב יוחננוב, מר איהב וותד
מתרגלים: יואב פרץ (מתרגל אחראי), אוראל אדיבי, נתי קולר, צביקה לזר, שלי גולן, שקד דרעי, רותם אליאס, תמיר שור, סער צור-שדי, אלעד גרוס, חן פרי, יהונתן לחמן, גיל אלגריסי.

סיסמה לפתיחת הצהרה על טוהר בחינות:

Insert_pwd

שאלה 1: [25 נקודות]

הנחיות ההגשה נמצאות בסוף השאלה.

סעיף א:

חשבו את סיבוכיות הזמן והמקום של הפונקציה $f1$ המוגדרת בקטע הקוד הבא, כפונקציה של n. n חיובי. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן.

```
void f1 (int n){
    int i = 0, count = 0;
    while (i < n) {
        if (i % 2 == 0) {
            count += i;
        }
        i = i * i + 3;
    }
}
```

סעיף ב:

חשבו את סיבוכיות הזמן והמקום של הפונקציה $f2$ המוגדרת בקטע הקוד הבא, כפונקציה של n. n חיובי. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן. ניתן להניח שסיבוכיות הזמן והמקום של printf היא $\theta(1)$.

```
void f2(int n){
    for (int i = n; i > 0; i /= 2){
        for(int j = i; j < n; j++){
            printf("*");
        }
    }
}
```

סעיף ג:

חשבו את סיבוכיות הזמן והמקום של הפונקציה $f3$ המוגדרת בקטע הקוד הבא, כפונקציה של n. n חיובי. אין צורך לפרט את שיקוליכם. חובה לפשט את הביטוי ככל שניתן.

```
void g3(int n, int sum) {
    if (n < 1) {
        printf("Sum: %d\n", sum);
        return;
    }
    g3(n - 1, sum + n);
    g3(n - 1, sum);
}
void f3(int n){
    g3(n, 0);
}
```

הנחיות ההגשה:

השתמשו בקובץ בשם exam_A1_q1.c הנתון לכם ונמצא במודל. הקובץ מכיל שלד לתוכנית המתאים למקרה שבו בכל השאלות בחרתם בתשובה "a".
מה שעליכם לעשות הוא לשנות את האות "a" לאות המתאימה לתשובה שלכם בכל סעיף. שימו לב שלכל סעיף יש לבחור גם את האות המתאימה לסיבוכיות הזמן "time complexity" וגם את האות המתאימה לסיבוכיות המקום "space complexity".
כך שההדפסות הסופיות למשל עבור סעיף א במידה ובחרתם "a" עבור סיבוכיות זמן ו "b" עבור סיבוכיות מקום ייראו כך:

printf("The time complexity of f1 is: %s\n" , "a");

printf("The space complexity of f1 is: %s\n" , "b");

אין לשנות שום דבר בקובץ השלד למעט האות אותה בחרתם לסעיף. לא יתקבלו ערעורים כתוצאה משינויים בקובץ השלד מעבר לאות שנבחרה.

איך בוחרים את האות המתאימה?

לפי המילון הנמצא בעמוד הבא.

[המשך השאלה בעמוד הבא]

a. $\theta(1)$

b. $\theta(loglog\ n)$

c. $\theta(log\ n)$

d. $\theta(\log^2 n)$

e. $\theta(\sqrt{n})$

f. $\theta(\sqrt{n}\log n)$

g. $\theta(\ n)$

h. $\theta(nlog\ n)$

i. $\theta(n\log^2 n)$

j. $\theta(n^{\frac{3}{2}})$

k. $\theta(n^2)$

l. $\theta(n^2\log(\log(n)))$

m. $\theta(n^2\log\ (n))$

n. $\theta(n^3 logn)$

o. $\theta(n^3)$

p. $\theta(n^4)$

q. $\theta\left(2^{\sqrt{n}}\right)$

r. $\theta\left(2^{\frac{n}{3}}\right)$

s. $\theta\left(2^{\frac{n}{2}}\right)$

t. $\theta(3^{\frac{n}{3}})$

u. $\theta\left(3^{\frac{n}{2}}\right)$

v. $\theta(2^n)$

w. $\theta(n * 2^n)$

x. $\theta(3^n)$

y. $\theta(n^2 3^n)$

z. $\theta(n^3 3^n)$

שאלה 2: [25 נקודות]

<u>הגדרה:</u> **מערך עולה-יורד מלא** הוא מערך שבו מתקיימות כל התכונות הבאות:

1. במערך קיים ערך מקסימלי.
2. כל הערכים שנמצאים משמאל לערך המקסימלי ממוינים בסדר עולה
3. כל הערכים שנמצאים מימין לערך המקסימלי ממוינים בסדר יורד.
4. ההפרש בערך מוחלט בין כל שני ערכים צמודים הוא **בדיוק** 1.

<u>הגדרה:</u> **מערך עולה-יורד חלקי** הוא מערך עולה יורד מלא, בלי התנאי האחרון.

ממשו את הפונקציה: $int\ exam_A1_q2(int\ arr[], int\ n, int\ k);$

המקבלת **מערך עולה-יורד חלקי**, ומחזירה את האיבר ה k **במערך עולה-יורד מלא** המכיל את המקסימום של arr ושהקצוות שלו הם הקצוות של arr.

<u>דוגמאות:</u>

ערכי הפרמטרים	פלט	הסבר
arr[] = {1, 3, 5, 2}, n = 4, k = 3	4	מערך עולה-יורד מלא שהערך המקסימלי בו הוא 5, האיבר הראשון בו הוא 1, והאחרון הוא 2, הוא: {1, 2, 3, 4, 5, 4, 3, 2} והערך במיקום 3 בו הוא 4
arr[] = {3, 4, 5, 3, 2, 1}, n = 6, k = 4	3	מערך עולה יורד מלא שהערך המקסימלי בו הוא 5, האיבר הראשון הוא 3 והאחרון הוא 1 הוא: {3, 4, 5, 4, 3, 2, 1} והאיבר במיקום 4 בו הוא 3.
arr[] = {1, 2, 3, 4}, n = 4 k = 2	3	מערך עולה יורד מלא שהערך המקסימלי בו הוא 4, הראשון הוא 1 והאחרון הוא 4 הוא: {1, 2, 3, 4} והאיבר במיקום 2 בו הוא 3.

[המשך השאלה בעמוד הבא]

הערות:

1. ב arr קיימים לפחות 2 איברים.
2. ניתן להניח כי הערך k חוקי, כלומר נמצא בגבולות המערך העולה-יורד מלא המתאים.
3. שימו לב שייתכנו חזרות, למשל בדוגמא השנייה, הערך 3 חזר גם בחלק העולה וגם בחלק היורד של המערך בקלט.
4. נתונה לכם תוכנית אשר קולטת את גודל המערך ואת איבריו, לאחר מכן קולטת את k, קוראת לפונקציה שהתבקשתם לממש ולבסוף מדפיסה את הערך שהפונקציה מחזירה.

דרישות סיבוכיות:

סיבוכיות זמן: $O(\log n)$ כאשר n הוא מספר האיברים ב arr.

סיבוכיות מקום נוסף: $O(1)$

שאלה 3: [25 נקודות]

<u>הגדרה</u>: בהנתן מערך arr בגודל n, ושלם חיובי k המערך arr יקרא k-**ממוין** אם מתקיים:

לכל $0 \le i < j < n$ $(arr[i]\%k) \le (arr[j]\%k)$

ממשו את הפונקציה: $void\ exam_A1_q3(int\ arr[\ ], int\ n, int\ k)$

1. המקבלת מערך של שלמים חיוביים *arr*, את גודלו *n* ופרמטר *k*. ומסדרת אותו מחדש כך שמתקיימים התנאים הבאים: המערך *arr* יהיה **מערך *k*-ממוין.**
2. אם לשני ערכים אותה שארית חלוקה ב *k*, אין חשיבות מי מהם יופיע קודם במערך *k*-ממוין.

<u>דוגמאות</u>:

קלט	המערך בסוף הפונקציה	הסבר
$arr[] = \{10, 30, 5, 2\}$ $n = 4$ $k = 3$	{30, 10, 5, 2} או {30, 10, 2, 5}	שארית החלוקה ב - 3 של איברי המערך: {1, 0, 2, 2} לכן 30 יופיע ראשון, אחריו 10 ולבסוף 5 ו – 2 בסדר כלשהו.
$arr[] = \{1,2,3,4,5,6,7,8,9\}$ $n = 9$ $k = 4$	פתרון אחד אפשרי: {4,8,1,5,9,2,6,3,7}	שארית החלוקה ב – 4 של איברי המערך: {1,2,3,0,1,2,3,0,1} לכן 4,8 יופיעו ראשונים, לא משנה באיזה סדר. 1,5,9 יופיעו אחריהם, לא משנה באיזה סדר. 2,6 יופיעו אחריהם, לא משנה באיזה סדר. וכנ"ל גם 3,7 יופיעו אחרונים, לא משנה באיזה סדר.

דרישות סיבוכיות:

סיבוכיות זמן: $O(n * k)$
סיבוכיות מקום נוסף: $O(1)$

הערות:

- מובטח שהמערך arr לא מכיל כפילויות
- נתונה לכם תוכנית אשר קולטת את גודל המערך ואת איבריו, לאחר מכן קולטת את k, קוראת לפונקציה שהתבקשתם לממש ולבסוף מדפיסה את ערכי המערך לאחר המיון.

שאלה 4: [25 נקודות]

בהינתן מטריצת מספרים שלמים mat בגודל $N \times N$ ואינדקסים (i,j) לתא מסוים במטריצה, נגדיר את ה**שכנים** של התא ה-(i,j):

- התא שמעליו $(i-1,j)$, אם הוא קיים.
- התא שמתחתיו $(i+1,j)$, אם הוא קיים.
- התא מימינו $(i,j+1)$, אם הוא קיים.
- התא משמאלו $(i,j-1)$, אם הוא קיים.

בנוסף, נגדיר את ה**צבע** של התא ה - (i,j) כערך הכתוב בו: $mat[i][j]$.

נגיד שהתא ה-(i,j) צבוע בצבע $c \in \mathbb{Z}$, אם $mat[i][j] == c$.

ממשו פונקציה שחתימתה:

$$void\ exam_A1_q4(int\ mat\ [N][N], int\ i, int\ j);$$

הפונקציה מקבלת מטריצת מספרים שלמים mat בגודל $N \times N$, אינדקסים i,j לתא מסוים במטריצה, ומשנה את המטריצה mat בצורה הבאה:

1. כותבת את הצבע 0 לתא ה-i,j.
2. כל השכנים של i,j שגם היו בצבע c שהיה ב $mat[i][j]$ במקור ייצבעו גם כן להיות 0.
3. כל השכנים של כל תא ששונה בעקבות כך ובאותו הצבע כמוהו גם כן ישנו את צבעם להיות 0. (וכן הלאה עד שאין יותר שינויים)

<u>דוגמת ריצה:</u>
עבור המטריצה mat הבא, ועבור $i=3, j=2$:

1	1	2	2	3
2	3	1	3	1
3	1	1	2	1
2	1	1	3	3
2	3	1	2	2

נקבל את רצף האירועים הבא:

הערך $mat[3][2]$ נצבע ל 0

השכנים של התא המקורי משמאלו מעליו ומתחתיו שבצבע 1 (כמו התא המקורי) יצבעו ל 0

השכנים שלהם שבצבע 1 גם יצבעו ל 0

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

⇦

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

⇦

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

וזה הערך הסופי של המטריצה mat.

הערות:

- יש לפתור את השאלה ברקורסיה בלבד – חל איסור להשתמש בלולאות. כלומר חל איסור מוחלט להשתמש במילים השמורות for, do ו־while.
  הערה זו נכונה הן לפונקציה אותה התבקשתם לממש והן עבור כל פונקציית עזר נוספת שתממשו בעצמכם.
- לרשותכם תוכנית הקולטת את המטריצה ואת i, j, קוראת לפונקציה ומדפיסה את המטריצה לאחר הצביעה שלה.
- ניתן להניח כי N מוגדר ב define.