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

מועד א׳ קיץ תשפ"ה

מבחן מסכם , 31.10.25

מספר סטודנט: | | | | | | | | | | *רשום/ה לקורס:* | 2 | 3 | 4 | 1 | 1 | |

משך המבחן: שלוש שעות.

הנחיות כלליות:

- **השימוש בכל חומר עזר אסור.**
- בדקו שיש X עמודים (4 שאלות) במבחן, כולל עמוד זה.
- 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 וגלובליים אלא אם נדרשתם לכך מפורשות.
- הקפידו על סגנון כתיבה כזה שקורא התוכנית ידע לפענח את רעיונותיכם:
  a. התוכנית יכולה להכיל תיעוד קל להבנה.
  b. על התוכנית להיות כתובה באופן מסודר ומדורג.
  c. יש לתת שמות פונקציות ושמות משתנים משמעותיים.
  d. ערכים קבועים יש להגדיר בעזרת define .
  e. ניתן להוסיף פונקציות עזר כרצונכם.
- 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.
- שימו לב שקיבלתם את קבצי ה C ובהם שלד של התוכנית. עליכם רק להשלים את המימוש של הפונקציה הנדרשת. אין לשנות את ה main.
- לכל שאלה קיבלתם מספר טסטים. מומלץ להוסיף טסטים שלכם.

המשך ההנחיות בעמוד הבא.

הנחיות הגשה: חשוב לקרוא בעיון!

1) לפני תחילת המבחן, יש למלא במודל הצהרה על טוהר הבחינות. סיסמה להצהרה במודל נמצאת בעמוד 3. ניתן למלא את ההצהרה רק בחצי השעה הראשונה.
2) הגשת כל שאלה במטלה נפרדת במערכת Gradescope רק לאחר ההצהרה על טוהר הבחינות.
3) יש למלא הצהרה על סיום הבחינה לאחר הגשת המטלה האחרונה שפתרתם ולפני היציאה מהכיתה.
4) במהלך המבחן מותר לעבוד אך ורק בסביבת CLion.
5) אין להשתמש בשום כלי בינה מלאכותית יוצרת כגון: CoPilot, ChatGPT וכדומה. שימוש בכלים אלו יגרור הליכים משמעתיים.
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, **תוביל לציון 0** והליכים משמעתיים.

נוסחאות שימושיות:

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

$$1^k+2^k+3^k+\ldots+n^k=\Theta(n^{k+1}) \qquad 1+k+k^2+k^3+\ldots+k^n=\Theta(k^n),\ k>1$$

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

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

$$\log_a(m\cdot n)=\log_a(m)+\log_a(n)$$
$$\log_a\left(\frac{m}{n}\right)=\log_a(m)-\log_a(n)$$
$$\log_a(m^n)=nlog_a(m)$$
$$n^{\log_n m}=m$$

צוות הקורס 234114/7

מרצים: גב' יעל ארז; **מתרגלת אחראית:** זהר נגל.

סיסמה להצהרה על טוהר הבחינות:

HomeSweetHome

שאלה 1: [25 נקודות]

ענו על השאלה במטלה המתאימה ב-Gradescope. לכל סעיף עם קטע קוד יופיעו שני תתי-סעיפים, אחד לסיבוכיות זמן ואחד לסיבוכיות מקום. בכל תת-סעיף עליכם לבחור אפשרות אחת ולסמן אותה. בכל הסעיפים יש להניח כי סיבוכיות הזמן והמקום של פונקציות הספרייה המוכרות $printf, free$ היא $\Theta(1)$. בנוסף יש להניח כי סיבוכיות הזמן של $malloc(n)$ היא $\Theta(1)$ וסיבוכיות מקום נוסף היא $\Theta(n)$.

א. חשבו את סיבוכיות הזמן והמקום של הפונקציה f1 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט את שיקוליכם. <u>חובה לפשט את הביטוי ככל שניתן</u>.

```
void f1(int n) {
    double** arr = malloc(n * sizeof(double*));
    if (!arr) return;
    for (int i = 1; i < n; ++i) {
        for (int j = 0; j < n; j += i) {
            arr[j] = malloc(j * sizeof(double));
            if (!arr[j]) return;
            for (int k = 0; k < j; k++) {
                arr[j][k] = 0;
            }
        }
        for (int j = 0; j < n; j += i) {
            free(arr[j]);
        }
    }
    free(arr);
}
```

ב. חשבו את סיבוכיות הזמן והמקום של הפונקציה f2 המוגדרת בקטע הקוד הבא, כפונקציה של n. אין צורך לפרט את שיקוליכם. <u>חובה לפשט את הביטוי ככל שניתן</u>.

```
int f2(int n) {
    if (n <= 1) return 0;
    int temp = n, counter = 0;
    while (temp > 1) {
        temp -= 2;
        counter++;
    }
    return f2(counter) + f2(counter);
}
```

ג. בקובץ finalA_q1.c תוכנית שבה משתמשים ב 2 מערכים: מערך ids של ת.ז. ומערך grades של ציונים. 2 המערכים באותו אורך, כאשר grades[i] הוא הציון הממוצע של הסטודנט שת.ז. שלו היא ids[i].
הת.ז. מיוצגת ע"י מחרוזת של 9 תווים בדיוק (סה"כ 10 בתים, יחד עם התו '\0'), והציון הוא מספר ממשי.

הפונקציה:

```
bool read_arrays(char*** ids, double** grades, int* n);
```

קוראת את המערכים מהמשתמש (אורך, ת.ז. וציון לכל סטודנט)

הפונקציה:

```
void free_arrays(char** ids, double* grades, int n);
```

משחררת את הזכרון שהוקצה באופן דינמי.

בקובץ יש גם פונקצית main אשר קוראת לפונקציה read_arrays, מדפיסה את הערכים (הציונים מודפסים בדיוק של 2 ספרות לאחר הנקודה), וקוראת לפונקציה free_arrays.

בקובץ יש טעויות. בפתרון עליכם להגיש את התוכנית המתוקנת.

לרשותכם הכנו בגריידסקופ טסט שנכשל עם התוכנית הנתונה.

שימו לב:

- אין לשנות את חתימות הפונקציות.
- זה שהטסט עובר לא אומר שתיקנתם את כל הטעויות.
- כל הקצאה דינמית יש לוודא שהצליחה.
- יש לשחרר כל זכרון שהוקצה דינמית.
- פונקציה המקבלת מצביע, תמיד צריכה לבדוק שהוא שונה מ NULL לפני שתשתמש בו.

שאלה 2: [25 נקודות]

עליכם לממש את הפונקציה הבאה:

$$int\ exam_q2(int\ arr[\ ], int\ n);$$

הפונקציה מקבלת מערך arr באורך n של מספרים טבעיים (שלמים גדולים מאפס).

המערך מייצג אזור עם n קירות. הקיר ה-i נמצא בנקודה ה-i בציר ה-x וגובהו $arr[i]$.

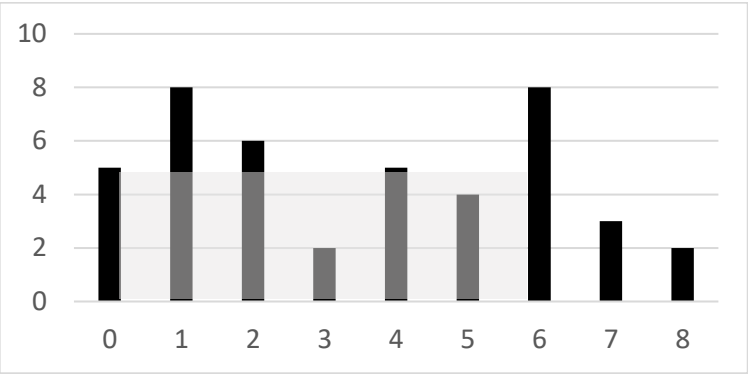
עליכם למצוא קיר שביחד עם הקיר השמאלי ביותר (הקיר במקום 0) וציר ה-x יוצרים את המיכל שיכול להכיל הכי הרבה מים. החזירו את כמות המים שמיכל זה מכיל.

דוגמאות:

קלט: $arr = \{5,8,6,2,5,4,8,3,2\}, n = 9$

פלט: 30

הסבר: מערך הגבהים המתואר בקלט נראה כך:



הקיר השמאלי של המיכל בגובה 5 (הקיר במקום 0). המיכל הגדול ביותר יתקבל אם נבחר את הקיר הימני להיות הקיר במקום 6. זהו מיכל ברוחב 6 ובגובה 5, כלומר כמות המים היא 5*6=30.

הערות:

- נתונה לכם תוכנית אשר קוראת מהמשתמש את אורך המערך, ולאחר מכן את תוכנו.
- ניתן להניח שהקלט תקין, בפרט arr מכיל רק מספרים חיוביים ו $n > 1$.

דרישות סיבוכיות:

- סיבוכיות זמן: $\Theta(n)$.
- סיבוכיות מקום: $\Theta(1)$.

שאלה 3 [25 נקודות]

מחרוזת t תקרא פרמוטציה של s אם t ו-s מכילות בדיוק את אותם תווים, אבל לא בהכרח באותו סדר.

למשל עבור המחרוזת "all" אוסף הפרמוטציות האפשריות (בסידור לקסיקוגרפי) הוא:

$$"all", "lal", "lla"$$

עליכם לממש את הפונקציה הבאה: $int\ exam_q3(char * s);$

הפונקציה מקבלת מחרוזת שמכילה אותיות קטנות באנגלית בלבד.

הפונקציה תבחן שתיים מהפרמוטציות של המחרוזת הנתונה: הקטנה והגדולה ביותר בסדר לקסיקוגרפי, ותחליף את המחרוזת באחת מבין שתיהן שיותר קרובה אליה (הכי מעט תווים שיש לשנות). אם שתי הפרמוטציות קרובות באותה מידע למחרוזת s, הפונקציה תחליף את s בפרמוטציה הקטנה. על הפונקציה להחזיר את מספר התווים ששונו ב-s.

דוגמאות

s לפני הקריאה לפונקציה	s אחרי הקריאה לפונקציה	ערך החזרה	הסבר
$dbcae$	$abcde$	2	הפרמוטציה הקטנה ביותר היא "$abcde$" והיא שונה מהמחרוזת המקורית ב 2 מקומות (יש להחליף את התו d בתו a ואת התו a בתו d) הפרמוטציה הגדולה ביותר היא "$edcba$" והיא שונה מהמחרוזת המקורית ב 4 מקומות (התו היחיד שנשאר במקום ולא מוחלף זה התו c)
$zxzyw$	$zzyxw$	3	הפרמוטציה הקטנה ביותר היא "$wxyzz$" והיא שונה מהמחרוזת המקורית ב 4 מקומות (התו היחיד שנשאר במקום ולא מוחלף זה התו x). הפרמוטציה הגדולה ביותר היא "$zzyxw$" והיא שונה מהמחרוזת המקורית ב 3 מקומות.
$abcd$	$abcd$	0	אין צורך לשנות את המחרוזת כיוון שהיא הפרמוטציה הקטנה ביותר והיא דורשת 0 שינויים.

הערות:

- נתונה לכם תוכנית אשר קוראת מהמשתמש את המחרוזת s (אורך ותוכן), קוראת לפונקציה ומדפיסה את הערך החוזר ואת המחרוזת החדשה.
- ניתן להניח שהקלט תקין (s מחרוזת חוקית המכילה רק אותיות קטנות).

דרישות סיבוכיות:

- סיבוכיות זמן: $\Theta(n)$ כאשר n הוא אורך המחרוזת s.
- סיבוכיות מקום: $\Theta(1)$.

שאלה 4: [25 נקודות]

ידוע כי בטקס מגדלי הירקות השנתי ישנה ארוחת ערב חגיגית בה מתיישבים הסועדים ב N שולחנות של 5 אנשים כל אחד. סה"כ מספר הסועדים הוא בדיוק 5N.

השפית הידועה מדאם כרובית אחראית על ארוחת הערב החגיגית ועליה לתכנן מראש אילו מנות להגיש לכל אחד מ N השולחנות. לשם כך יש להתחשב באילוצים הבאים:

- כל סועד סימן מראש שלוש מנות מתוך תפריט של K מנות, כאשר לכל מנה יש מספר שונה החל מ-0 ועד K-1. יש להכין לו בדיוק מנה אחת מתוך השלוש שסימן.
- משיקולים קולינריים, ישנן מנות שלא יכולות להיות יחד באותו שולחן. אילוצים אלו מתוארים במטריצת אילוצים בגודל KxK, בה האיבר במקום i,j הוא 0 אם מנה i לא יכולה להיות עם מנה j באותו שולחן, ו 1 אם הן יכולות להיות יחד.

עליכם לממש את הפונקציה הבאה: $int\ exam_q4(int\ mat1[\quad][3], int\ mat2[K][K])$

הפונקציה מקבלת את בחירות הסועדים במטריצה $mat1$ (שורות 0 עד 4 הסועדים בשולחן הראשון, שורות 5 עד 9 הסועדים בשולחן השני, וכך הלאה), ואת מטריצת האילוצים $mat2$.
על הפונקציה להחזיר את מספר המנות השונות המינימלי שיכול לקיים את כל האילוצים. אם לא קיימת אפשרות העונה על כל האילוצים יש להחזיר 1-.
למשל, עבור 2 שולחנות ו-6 מנות, אם בחירת הסועדים היא:

$$\begin{aligned} mat1 = \{&\{0,1,2\},\\ &\{0,1,2\},\\ &\{0,1,2\},\\ &\{0,1,2\},\\ &\{0,3,5\},\\ &\{3,4,5\},\\ &\{3,4,5\},\\ &\{3,4,5\},\\ &\{3,4,5\},\\ &\{0,3,5\}\} \end{aligned}$$

ומטריצת האילוצים היא:

$$\begin{aligned} mat2 = \{&\{1,1,1,1,1,0\},\\ &\{1,1,1,1,1,1\},\\ &\{1,1,1,1,1,1\},\\ &\{1,1,1,1,1,1\},\\ &\{1,1,1,1,1,1\},\\ &\{0,1,1,1,1,1\}\} \end{aligned}$$

כלומר כל המנות יכולות להיות על אותו שולחן, מלבד מנות 0 ו-5 שלא יכולות להיות על אותו שולחן.

המשך השאלה בעמוד הבא!

ניתן להגיש לכל הסועדים בשולחן הראשון את מנה 0, ובשולחן השני ניתן להגיש לכולם את מנה 3.

לכן הפונקציה תחזיר 2 (2 מנות שונות יוגשו לשולחנות).

<u>הערות:</u>

- נתונה לכם תוכנית אשר קוראת מהמשתמש את בחירות הסועדים ומטריצת האילוצים.
- יש להשתמש בשיטת `backtracking` כפי שנלמדה בכיתה.
- בשאלה זו אין דרישות סיבוכיות, אולם כמקובל ב-`backtracking` יש לוודא שלא מתבצעות קריאות רקורסיביות מיותרות עם פתרונות שאינם חוקיים.
- ניתן ומומלץ להשתמש בפונקציות עזר (ויש לממש את כולן).
- ניתן להניח כי מטריצת האילוצים סימטרית ואיברי האלכסון כולם 1.

בהצלחה!