

מבוא למדעי המחשב מ'ח' (234114/7), סמסטר קיץ 2018

תרגיל בית 5

מועד אחרון להגשה: יום שלישי 17.09.18

המתרגל האחראי על תרגיל זה: עמית אליהו

משרד: טאוב 320

E-mail: amiteliyahu10@gmail.com

הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרון.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל, עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי על התרגיל לפני תאריך הגשת התרגיל.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
 - השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
 - וודאו את האותיות הגדולות והקטנות לפי הדוגמאות וההסברים בתרגיל.
 - אין להדפיס רווחים שלא התבקשתם להדפיס (בתחילת שורה או בסופה).
 - השתמשו באתר הבדיקה העצמית.
- בתרגיל זה מותר להשתמש בפונקציות מהספריית **stdbool.h, stdio.h, stdlib.h** בלבד, שנלמדו בהרצאות ובתרגולים, אלא אם צוין אחרת.
- ההגשה הינה אלקטרונית וב**בודדים** דרך אתר הקורס. קובץ ההגשה יהיה מסוג **zip** (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
 - קובץ **students.txt** עם מספר תעודת הזהות שלך וכתובת האי-מייל שלך.
 - קובץ פתרון **hw5q1.c** לשאלה 1
 - קובץ פתרון **hw5q2.c** לשאלה 2
 - קובץ פתרון **hw5q3.c** לשאלה 3
- **חובה לשמור את קוד אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!** אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

שאלה 1 - מיונים

אינטרוול הוא קטע המוגדר על-ידי נקודת התחלה ונקודת סיום, למשל (3,5) הוא אינטרוול המתחיל בנקודה 3 ומסתיים בנקודה 5. בנוסף נקודת הסיום תמיד גדולה ממש מנקודת ההתחלה. על מנת למיין אינטרוולים, נגדיר יחס סדר עבור שני אינטרוולים (w,x) , (y,z) באופן הבא:

1. אם אחד מתחיל לפני השני אז הוא יהיה קודם לו בסדר, כלומר אם $w < y$ אז האינטרוול (w,x) קודם לאינטרוול (y,z) .
2. אם נקודת ההתחלה זהה, האינטרוול שמסתיים קודם יהיה קודם לאינטרוול שמסתיים מאוחר יותר אם $w=y$ אבל $x < z$ אז האינטרוול (w,x) קודם לאינטרוול (y,z) .

דוגמה

סדרת אינטרוולים ממויינת: (1,6), (2,12), (4,9), (4,11)

סדרת אינטרוולים **שאינה** ממויינת: (1,6), (4,9), (3,17)

ממשו את הפונקציה הבאה:

```
void sort_intervals(int** intervals, int n);
```

intervals הוא מערך דו-ממדי בגודל $n \times \text{DIM}$ (n שורות, DIM עמודות) כאשר $\text{DIM}=2$ מוגדר ב `define`, כל שורה מייצגת אינטרוול כאשר העמודה הראשונה מייצגת התחלה והעמודה השנייה מייצגת סיום, המערך עבור הסדרה השנייה של אינטרוולים יראה:

| | |
|---|----|
| 1 | 6 |
| 4 | 9 |
| 3 | 17 |

הפונקציה ממיינת את intervals לפי הסדר שהגדרנו. לאחר הרצת הפונקציה `sort_intervals` על מערך זה, המערך יהיה ממויין:

| | |
|---|----|
| 1 | 6 |
| 3 | 17 |
| 4 | 9 |

סיבוכיות זמן נדרשת: $O(n \log n)$
סיבוכיות מקום נדרשת: $O(n)$

תוכנית ראשית

1. מדפיסים את ההודעה הבאה

Please enter the number of intervals:

2. קולטים מהמשתמש מספר חיובי k .
3. קולטים סדרת אינטרוולים לפי הפורמט (יוסבר בהמשך) עד קבלת k אינטרוולים או הגעה ל EOF.
4. מדפיסים את רשימת האינטרוולים הממוינת לפי הפורמט.

פורמט קליטה/הדפסה של אינטרוולים

1. כל אינטרוול ייוצג על ידי ערך התחלה רווח ערך סיום.
2. בין כל שני אינטרוולים יש ירידת שורה.

למשל הסדרה מקודם תראה:

| | |
|---|----|
| 1 | 6 |
| 4 | 9 |
| 3 | 17 |

הערות

1. כל האינטרוולים חוקיים כלומר בכל שורה יהיה שני ערכים, ערך התחלה וערך סיום כאשר ערך הסיום גדול ממש מערך ההתחלה. ניתן להניח שהקלט לא יסתיים באמצע שורה.
2. כל האינטרוולים שונים כלומר אין שני אינטרוולים עם אותם נקודות התחלה וסיום.
3. אפשר להניח שהמספר k חיובי.
4. צריך לדאוג שהפונקציה `sort_intervals` תקבל את המערך באורך שנקלט בפועל, כלומר אם $k=1000$ והמשתמש הכניס רק 10 אינטרוולים (עם EOF בסוף) אז לשלוח מערך בגודל 10 ולא 1000 ל-`sort_intervals`.
5. אפשר לעשות הקצאה דינאמית של מערך דו-ממדי בגודל $m \times n$ (נ שורות ו m עמודות) באופן הבא:

```
int** matrix = (int**) malloc(sizeof(int*)*n);
for (int i=0;i<n;i++) {
    matrix[i] = (int*)malloc(sizeof(int)*m);
}
```

הסבר: מקצים מערך של מצבעי `int` בגודל n ואז לכל איבר מקצים מערך של `int` בגודל m .

משחררים מערך זה באופן הבא:

```
for (int i=0;i<n;i++) {
    free(matrix[i]);
}
free(matrix);
```

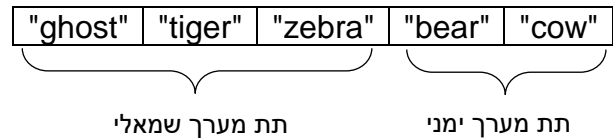
שאלה 2 - רקורסיה

מערך מחרוזות **ממוין** הוא מערך בו המחרוזות מסודרות בסדר לקסיקוגרפי עולה (לפי ערכי ASCII).

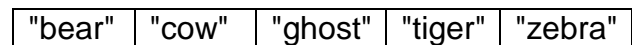
מערך מחרוזות **ממוין ציקלית** מקיים את שני התנאים הבאים:

- המערך הוא שרשור של שני תתי מערכים ממוינים (אחד מהם יכול להיות ריק).
- כל המילים בתת המערך הימני קטנות (מבחינת סידור לקסיקוגרפי) מכל המילים בתת המערך השמאלי.

למשל המערך הבא הוא מערך ממוין ציקלית:



שימו לב שגם מערך ממוין הוא מערך ממוין ציקלית:



במקרה זה אחד מתתי המערכים הוא בגודל אפס.
עליכם לממש את הפונקציה:

```
int find_min(char *arr[], int n);
```

אשר מקבלת מערך מחרוזות `arr` **ממוין ציקלית** ואת אורכו `n`, ומחזירה את האינדקס של **המחרוזת הקטנה ביותר** (כלומר המחרוזת שמופיעה ראשונה במילון).

מותר ואף מומלץ לממש פונקציית `strcmp` שראינו בתרגול 9. פונקציה זו יכולה להכיל לולאות.

שאר המימוש צריך להיות **רקורסיבי ובלתי לולאות (חוץ מקריאה ל `strcmp`)**.

הניחו שאורך המחרוזת הארוכה ביותר לא יעבור את `M` כאשר `M=100` מוגדר ב `define`.

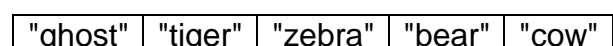
סיבוכיות זמן נדרשת: $O(M \log n)$.

סיבוכיות מקום נדרשת: $O(\log n)$.

* **הסיבוכיות נוגעת לצריכת המשאבים אותם צורכת הפונק' בלבד.**

דוגמאות

עבור המערך הבא:



הפונקציה תחזיר 3, כיוון שהמילה "bear" מופיעה ראשונה במילון.

עבור המערך הבא:

| | | | | |
|--------|-------|---------|---------|---------|
| "bear" | "cow" | "ghost" | "tiger" | "zebra" |
|--------|-------|---------|---------|---------|

הפונקציה תחזיר 0, כיוון ששוב המילה "bear" מופיעה ראשונה במילון.

תוכנית ראשית

1. מדפיסים את ההודעה הבאה

Please enter the number of words:

2. קולטים מהמשתמש מספר חיובי n.
3. קולטים מהמשתמש n מחרוזות כאשר יש ירידת שורה בין כל שתי מחרוזות, המחרוזות יהוו מערך מחרוזות ממין ציקלית.
4. מדפיסים את אינדקס המחרוזת הקטנה ביותר.

הערות

- ניתן להניח שאין במערך 2 מילים זהות.
- אסור לשנות את המחרוזות אפילו לא באופן זמני.
- ניתן להניח שהמחרוזות מורכבות מאותיות אנגליות קטנות בלבד.
- ניתן להניח שהמשתמש מכניס בדיוק n מחרוזות.

שאלה 3 - Backtracking

בהינתן סדרת מספרים s נגדיר את סדרת ההפרשים של s להכיל את ההפרשים של כל שני איברים עוקבים ב s . למשל אם $s=\{10,15,9,100,70\}$ אזי סדרת ההפרשים $\{5,-6,91,-30\}$.

סדרת מספרים **תקרא משתוללת הפרשים** אם סדרת ההפרשים שלה מחליפה סימן, כלומר האיבר הראשון בסדרת ההפרשים יכול להיות חיובי או שלילי ושאר האיברים מקיימים שסימן כל איבר הפוך מסימן האיבר שלפניו.

ערך ההשתוללות של סדרה הוא אורך תת-הסדרה הארוכה ביותר שמשתוללת הפרשים.
שימו לב:

1. בתרגיל זה תת-סדרה אינה בהכרח מופיעה ברצף במערך.
2. יתכן שקיימות כמה תתי-סדרות משתוללות באורך מקסימלי.

דוגמה

עבור הסדרה $\{9,8,7,6,11\}$ ערך ההשתוללות הוא 3, כי תת-הסדרה $\{9,8,11\}$ משתוללת הפרשים (גם $\{8,6,11\}$ משתוללת הפרשים למשל) אבל אין תת-סדרה באורך 4 או יותר המשתוללת הפרשים.

ממשו את הפונקציה `max_crazy` המקבלת מערך חד ממדי `series` באורך n ומחזירה את ערך ההשתוללות של הסדרה שמייצג.

```
int max_crazy(int series[], int n);
```

תוכנית ראשית

1. מדפיסים את ההודעה הבאה.

Please enter the length of the sequence:

2. קולטים מהמשתמש מספר חיובי n .
3. קולטים מהמשתמש n מספרים כאשר יש ירידת שורה בין כל שני מספרים.
4. מדפיסים ערך ההשתוללות של סדרת המספרים שהכניס המשתמש.

הערות

1. אפשר להניח שכל האיברים שונים במערך.
2. סדרה באורך 2 משתוללת הפרשים באופן ריק.
3. ניתן להניח שהמשתמש יכניס בדיוק n ערכים.
4. ניתן להניח שהסדרה לפחות באורך 2.