

מבוא למדעי המחשב מ'ח' (234114/7), סמסטר אביב 2021

תרגיל בית 5

מועד אחרון להגשה: 01.07.2021 23:59

- המתרגל האחראי על תרגיל זה: ירון חי
- שעות קבלה מוגדרות עבור התרגיל:
 - יום רביעי ה-23 ביוני בין השעות 9:30-10:30
 - יום שני ה-28 ביוני, בין השעות 12:30-1:30
- שעות הקבלה יתקיימו בבניין טאוב חדר 520. ניתן גם לקיימן בזום, בתיאום מראש במייל.
- דוא"ל לנושאים אישיים בלבד בנושא התרגיל: aron.hay+234114#2021hw5@cs.technion.ac.il
- דוא"ל בנושא תרגיל זה שלא ישלח לכתובת זו, לא יזכה למענה.
- מלבד נושאים אישיים, יש לשאול שאלות בנושא התרגיל בפיאצה בלבד.

הנחיות כלליות

- הגשה בבודדים. 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.
- הבהרות שפורסמו באתר הפיאצה הן מחייבות, ויש לראות בהן כאילו היו כתובות בהוראות לתרגיל זה.

מדיניות ערעורים, הארכות ובדיקה

- 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 (פונקציות הדפסה סופקו עם התרגיל) ובדיקו 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).
- בכל סוף שורה יש להדפיס תו ירידת שורה, גם בשורה האחרונה.
- השתמשו באתר הבדיקה העצמית.

ספריות מותרות לשימוש

בתרגיל זה מותר להשתמש בפונקציות מהספריות הבאות **בלבד**, שנלמדו בהרצאות ובתרגולים, אלא אם צוין אחרת:

- `stdio.h`
- `stdbool.h`
- `stdlib.h`

החומר הנדרש לתרגיל זה שייך לתרגולים 1-13. אין להשתמש בחומר שאינו מופיע במצגות אלה.

אופן ההגשה

- ההגשה הינה אלקטרונית ו**בבודדים** 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.

שאלות לגבי התרגיל או פניות בנושא התרגיל

1. בכל שאלה בנושא התרגיל:
 - a. יש לבדוק שלא קיימת שאלה זזה (או דומה), שתשובה עליה תענה על שאלתכם.
 - b. אם אין שאלה דומה, יש לפרסם את השאלה שלכם באתר הפיאצה (ולחכות למענה).
2. פניות בנושאים אישיים יש לשלוח אל כתובת הדוא"ל המופיעה בעמוד הקודם.
- a. למען הסר ספק, שאלה בנושאי הגדרות התרגיל או דרישות התרגיל איננה פניה בנושא אישי.
3. פניות שיתקבלו שלא לפי ההנחיות האלה, לא יענו.

ב ה צ ל ח ה !!

רקורסיה (שאלה 1)

כתבו תוכנית רקורסיבית הקולטת מערך של מספרים שלמים שונים ואת גודלו, ומדפיסה את המספר השני הכי גדול במערך. על התכנית לפעול בסיבוכיות זמן של $O(n)$ (לינארי באורך המערך n).

דוגמות הרצה

```
Please enter length:
3
Please enter array:
3 5 1
The second-greatest is:
3
```

```
Please enter length:
4
Please enter array:
3 5 1 9
The second-greatest is:
5
```

פקודות ההדפסה לנוחיותכם

```
printf("Please enter length:\n");
printf("Please enter array:\n");
printf("The second-greatest is:\n%d", <value>);
```

Backtracking (שאלה 2)

יוג'י משוטט ברחבי מדינה בת N ערים. הוא נעזר במטריצת כבישים $N \times N$, בה התא (i,j) מכיל את אורך הכביש המחבר בין העיר i לעיר j . זוהי מטריצה בה איברי האלכסון הראשי שווים 0, שאר האיברים חיוביים ממש, והיא אינה בהכרח סימטרית (ייתכנו כבישים חד-סטריים). יוג'י שם לב שהכביש הישיר בין שתי ערים אינו בהכרח הדרך הקצרה ביותר לעבור ביניהן, כלומר לעתים משתלם לעבור דרך ערים אחרות.

כתבו תוכנית הקולטת:

1. מטריצה $N \times N$ עבור $N=4$ (יש להגדיר זאת ב-define)
2. עיר מקור
3. עיר יעד (מובטח כי היא שונה מעיר המקור)

על התוכנית להדפיס עבור יוג'י את המסלול הקצר ביותר מהמקור ליעד (כולל המקור והיעד).

יש לפתור באמצעות טכניקת backtracking שנלמדה במסגרת הקורס. למרות שאין דרישות סיבוכיות מפורשות לשאלה זו, עליכם להימנע מסריקת פתרונות מיותרים (קריאות רקורסיביות מיותרות) שניתן לחסוך כפי שנלמד בקורס. יש למנוע את סריקתם מוקדם ככל האפשר.

הערות

1. מובטח כי קיים מסלול קצר ביותר יחיד בין המקור ליעד.
2. ניתן להניח שהקלט תקין, ואין צורך לבדוק זאת.

דוגמאות הרצה

```
Please enter roads matrix:
0 5 2 2
1 0 1 1
1 2 0 1
1 1 2 0
Please enter source city:
0
Please enter destination city:
1
The shortest path is:
0 3 1
```

```
Please enter roads matrix:
0 5 3 2
1 0 2 1
1 2 0 1
1 1 2 0
Please enter source city:
3
Please enter destination city:
2
The shortest path is:
3 2
```

פקודות ההדפסה לנוחיותכם

```
#define N 4
printf("Please enter roads matrix:\n");
printf("Please enter source city:\n");
printf("Please enter destination city:\n");

void print_shortest_path(int* sol, int n) {
    printf("The shortest path is:\n");
    while(int i = 0; i < n - 2; i++) {
        printf("%d ", sol[i]);
    }
    printf("%d", sol[n - 1]);
}
```

מיונים (שאלה 3)

כתבו תוכנית הקולטת:

1. מערך arr של מספרים שלמים שונים
2. את אורכו n
3. מספר שלם x

על התכנית להדפיסה האם קיימים במערך 3 איברים שונים שסכומם x.
על התוכנית לפעול בסיבוכיות זמן של $O(n^2)$.

פקודות ההדפסה לנוחיותכם

```
printf("Please enter length:\n");  
printf("Please enter array:\n");  
printf("Please enter sum:\n");  
printf("Such elements do exist.");  
printf("Such elements do not exist.");
```

דוגמאות הרצה

```
Please enter length:  
4  
Please enter array:  
1 2 5 3  
Please enter sum:  
6  
Such elements do  
exist.
```

```
Please enter length:  
4  
Please enter array:  
1 2 5 3  
Please enter sum:  
5  
Such elements do not  
exist.
```

ב ה צ ל ח ה !!