

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 1
מספר השאלון: 899271
שם התלמיד/ה: _____

מדעי המחשב

מבני נתונים ומסלול - 4-5 יחידות לימוד

הוראות

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

פרק ראשון - 2×25 - 50 נקודות

פרק שני - 2×25 - 50 נקודות

סך הכול - 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

ד. הוראות מיוחדות:

(1) רשמו על הכריכה החיצונית של המחברת את שם המסלול שלמדתם.

המסלול הוא אחד משלושת המסלולים האלה: אלגוריתמים, מודלים חישוביים, תכנות מונחה עצמים.

(2) את כל התוכניות שיש לכתוב בשפת מחשב בפרקים הראשון והשני כתבו בשפה אחת בלבד - Java או C#.

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.

כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

בשאלון זה שני פרקים. יש לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה:

`Scanner input = new Scanner(System.in);`

שימו לב: בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ בינרי וחוליה,

בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

השאלות

פרק ראשון

ענו על שתיים מן השאלות 1-3 (לכל שאלה - 25 נקודות).

1. (25 נקודות)

שימו לב: בשאלה זו אפשר להשתמש בפעולות של המחלקות תור, עץ בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן. שני איברים בתור של מספרים, הנמצאים במקומות שונים בתור, הם "זוג קרוב" לפי d אם ההפרש בין ערכיהם (בערך מוחלט) קטן מ- d או שווה לו. ממשו את הפעולה החיצונית שלפניכם:

Java - `public static int closePairs (Queue<Integer> q, int d)`

#C - `public static int ClosePairs (Queue<int> q, int d)`

הפעולה מקבלת תור q של מספרים שלמים ומספר שלם d הגדול מ-0 או שווה לו, ומחזירה את מספר הזוגות הקרובים לפי d שיש בתור q . כל זוג נספר פעם אחת בלבד. דוגמה: עבור התור q שלפניכם ו- $d = 2$ הפעולה תחזיר 4, כי בתור יש ארבעה זוגות קרובים: (8, 9), (3, 4), (5, 4), (4, 5).

	ראש התור							
q	8	3	12	5	9	20	4	

עבור אותו התור ו- $d = 0$ הפעולה תחזיר 0 (אין בתור שני מספרים שווים).

הערות:

- הניחו שבתור q יש שני איברים לפחות.
- אין צורך לשמור על התור q .
- אין להשתמש בשאלה זו במערך וברשימה מקושרת.
- ממשו את הפעולה `closePairs/ClosePairs` המתוארת לעיל.

2. (25 נקודות)

נוסח Java

שימו לב: לשאלה זו שני נוסחים: בשפת Java ובשפת #C.

לפותרים בשפת Java

א. (1) נתונה הפעולה `what`:

Java

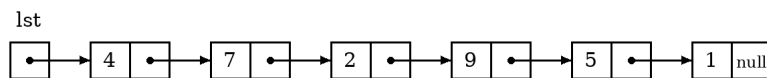
```
public static int what (Node<Integer> lst, int x)
{
    if (lst == null)
        return 0;
    int r = what (lst.getNext(), x);
```

```

if (lst.getValue() < x)
    return r + 1;
return r;
}

```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה `what (lst, 5)`, ורשמו את הערך שהפעולה מחזירה.

(2) מה עושה הפעולה `what`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `what`? נמקו את תשובתכם.

ב. (1) נתונה הפעולה `guess`:

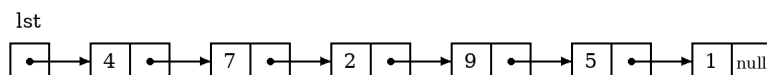
Java

```

public static int guess (Node<Integer> lst)
{
    if (lst == null)
        return 0;
    return what (lst.getNext(), lst.getValue()) + guess (lst.getNext());
}

```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה `guess (lst)`, ורשמו את הערך שהפעולה מחזירה.

בסעיף זה אין צורך לעקוב אחר הפעולה `what`.

(2) מה עושה הפעולה `guess`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `guess`? נמקו את תשובתכם.

נוסח C#

שימו לב: לשאלה זו שני נוסחים: בשפת Java ובשפת C#.

לפותרים בשפת C#

א. (1) נתונה הפעולה `What`:

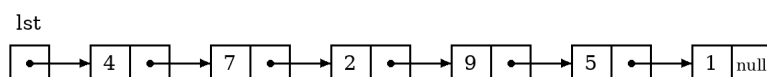
C#

```

public static int What (Node<int> lst, int x)
{
    if (lst == null)
        return 0;
    int r = What (lst.GetNext(), x);
    if (lst.GetValue() < x)
        return r + 1;
    return r;
}

```

נתונה שרשרת חוליות - lst מטיפוס שלם:



(1) עקבו אחר הפעולה `What (lst, 5)`, ורשמו את הערך שהפעולה מחזירה.

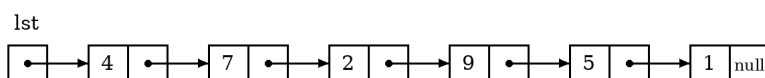
(2) מה עושה הפעולה `What`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `What`? נמקו את תשובתכם.

ב. (1) נתונה הפעולה `Guess`:

```
C#
public static int Guess (Node<int> lst)
{
    if (lst == null)
        return 0;
    return What (lst.GetNext(), lst.GetValue()) + Guess (lst.GetNext());
}
```

נתונה שרשרת חוליות - `lst` מטיפוס שלם:



(1) עקבו אחר הפעולה `Guess (lst)`, ורשמו את הערך שהפעולה מחזירה.

בסעיף זה אין צורך לעקוב אחר הפעולה `What`.

(2) מה עושה הפעולה `Guess`? הסבירו את תשובתכם.

(3) מהי סיבוכיות הפעולה `Guess`? נמקו את תשובתכם.

3. (25 נקודות)

שימו לב: בשאלה זו אפשר להשתמש בפעולות של המחלקות `Tor`, `ץ` בינרי וחוליה, בלי לממש אותן. אם משתמשים בפעולות נוספות, יש לממש אותן.

לפניכם הפעולה החיצונית `numDigits/NumDigits`. אפשר להשתמש בפעולה בלי לממש אותה.

תיאור הפעולה	כותרת הפעולה
הפעולה מקבלת מספר שלם <code>num</code> הגדול מ-0 או שווה לו, ומחזירה את מספר הספרות שלו.	<code>public static int numDigits - Java (int num)</code>
לדוגמה: עבור 4825 הפעולה מחזירה 4, ועבור 0 היא מחזירה 1.	<code>public static int NumDigits - #C (int num)</code>

א. ממשו את הפעולה החיצונית שלפניכם:

`public static boolean cutDigits (BinNode<Integer> tr) - Java`

`public static bool CutDigits (BinNode<int> tr) - #C`

הפעולה מקבלת צומת ללא בנים (עלה) שערכו מספר שלם הגדול מ-0 או שווה לו. נסמן ב-`d` את מספר הספרות של ערך הצומת.

אם $d = 1$ (מספר חד-ספרתי), הפעולה מחזירה `false`.

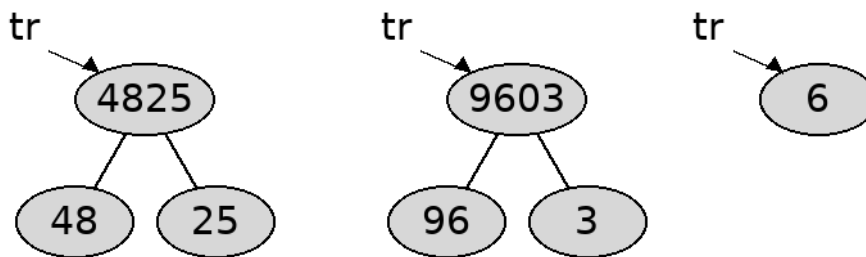
אחרת, הפעולה "חותכת" את המספר לשני חלקים ומוסיפה לצומת שני בנים: ערך הבן הימני הוא

המספר שיוצרות $d/2$ הספרות הימניות של ערך הצומת (חלוקה בשלמים), וערך הבן השמאלי הוא

המספר שיוצרות שאר הספרות (השמאליות). אם החלק הימני מתחיל בספרה 0, האפסים

שבתחילתו אינם נשמרים (למשל, 03 נשמר כ-3). לאחר ההוספה הפעולה מחזירה `true`.

דוגמאות: בתום הפעולה ייראו הצמתים כך (עבור הערכים 4825, 9603, 6):



(עבור $d = 3$: 307, ולכן הבן הימני הוא 7 - ספרה אחת - והבן השמאלי הוא 30.)

ב. (1) נתונה הפעולה what/What שלפניכם, המקבלת צומת ללא בנים שערכו מספר שלם חיובי **שאין בו את הספרה 0**.

בשפת:

Java

```
public static int what(BinNode<Integer> tr) {
    if (!cutDigits(tr))
        return 1;
    return what(tr.getLeft()) + what(tr.getRight());
}
```

C#

```
public static int What(BinNode<int> tr)
{
    if (!CutDigits(tr))
        return 1;
    return What(tr.GetLeft()) + What(tr.GetRight());
}
```

(1) סרטטו את העץ כפי שהוא יראה בתום הפעולה what/What עבור צומת ללא בנים - tr

שערכו 73846, וכתבו מה הפעולה מחזירה. יש להראות מעקב.

(2) הסבירו מה מבצעת הפעולה what/What ומה היא מחזירה.

פרק שני (50 נקודות) - אלגוריתמים

יש לענות על **שתי** שאלות במסלול שלמדתם (לכל שאלה - 25 נקודות).

4. (25 נקודות)

לפניכם שני סעיפים, א-ב, שאין קשר ביניהם. ענו על **שני** הסעיפים.

א. לפניכם שבע טענות (1)-(7). בחרו **בחמש** מהן, וכתבו את מספריהן. ציינו בנוגע לכל טענה שבחרתם

אם היא נכונה או לא נכונה. אם הטענה נכונה - נמקו מדוע, ואם היא לא נכונה - הביאו דוגמה

נגדית מגרף שיש בו **ארבעה קודקודים לפחות**.

(1) בכל גרף לא מכוון, מספר הקודקודים שדרגתם אי-זוגית הוא מספר זוגי.

(2) נתון גרף קשיר לא מכוון שיש בו n קודקודים ו- $1 - n$ קשתות. אין בגרף מעגל.

(3) נתון גרף ממושקל קשיר לא מכוון, ובו קשת e שמשקלה **גדול** ממשקל כל אחת משאר הקשתות.

הקשת e אינה נמצאת באף עץ פורש מינימלי של הגרף.

(4) נתון גרף ממושקל קשיר לא מכוון, ובו קשת e שמשקלה **קטן** ממשקל כל אחת משאר הקשתות. הקשת e נמצאת בכל עץ פורש מינימלי של הגרף.

(5) נתון גרף מכוון G . אם סריקת DFS מקודקוד s מגלה את כל הקודקודים בגרף, אז הגרף קשיר היטב.

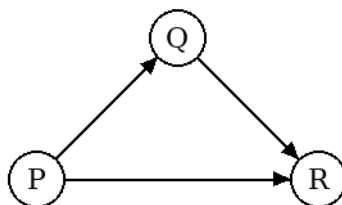
(6) בגרף מכוון חסר מעגלים (גמ"ל) שיש בו קשתות במשקל שלילי, האלגוריתם למציאת מסלולים קצרים בגמ"ל (מעבר על הקודקודים לפי מיון טופולוגי ושיפור הקשתות היוצאות מכל קודקוד) מחשב נכון את המרחקים הקצרים (הקלים) מקודקוד המקור.

(7) נתון גרף ממושקל קשיר לא מכוון שבו לכל קשת משקל שונה. עץ המסלולים הקצרים (הקלים) מקודקוד s , שמתקבל באלגוריתם של דייקסטרה, הוא תמיד עץ פורש מינימלי של הגרף.

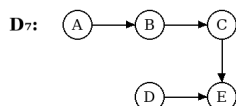
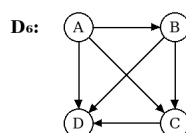
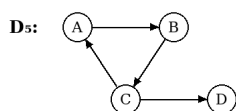
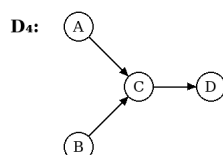
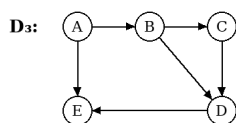
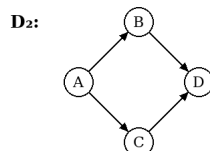
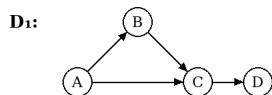
ב. **מיון טופולוגי** של גרף מכוון הוא סידור של כל קודקודי הגרף בשורה, כך שלכל קשת $u \rightarrow v$ הקודקוד u מופיע לפני הקודקוד v .

גרף מכוון מכונה "**מיון יחיד**" אם יש לו **בדיוק** מיון טופולוגי אחד.

דוגמה: הגרף שלפניכם הוא גרף "**מיון יחיד**" - המיון הטופולוגי היחיד שלו הוא P, Q, R .



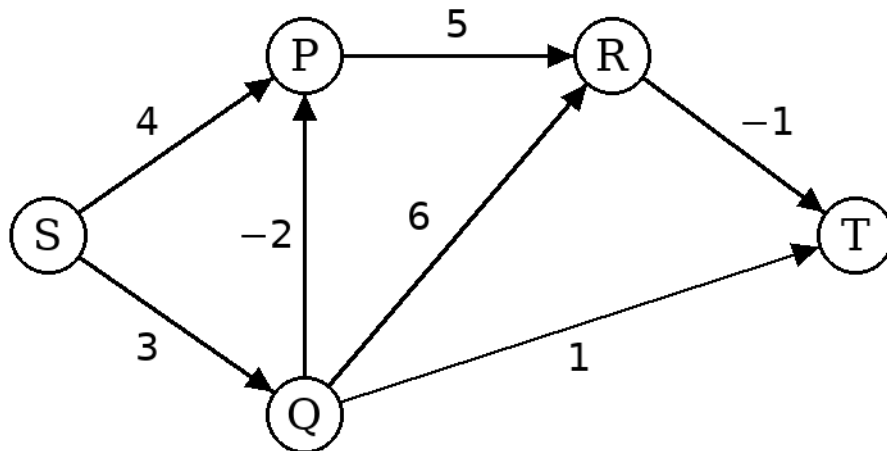
לפניכם שבעה גרפים מכוונים D_1 – D_7 . בחרו בארבעה מהם, וכתבו בנוגע לכל גרף שבחרתם אם הוא גרף "**מיון יחיד**" או לא. אם כתבתם שלא - הציגו שני מיונים טופולוגיים שונים של הגרף, או הסבירו מדוע אין לגרף מיון טופולוגי כלל.



5. (25 נקודות)

בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענו על שני הסעיפים.

א. (1) לפניכם גרף מכוון ממושקל G_i . בגרף יש קשתות שמשקלן שלילי, אך אין בו מעגל שמשקלו שלילי:



(1) איזה אלגוריתם מחזיר את המסלול הקצר (הקל) בין שני קודקודים בגרף מכוון ממושקל **כלשהו**, שבו ייתכנו משקלים **שליליים** (אך אין מעגל שמשקלו שלילי)? מהי סיבוכיות זמן הריצה של האלגוריתם? נמקו את תשובתכם.
נוסף על כך, הריצו את האלגוריתם שכתבתם, בעבור הגרף הנתון G_i , מקודקוד S לקודקוד T. בכל סבב עברו על הקשתות בסדר הזה:

$$R \rightarrow T, P \rightarrow R, Q \rightarrow P, Q \rightarrow R, Q \rightarrow T, S \rightarrow P, S \rightarrow Q$$

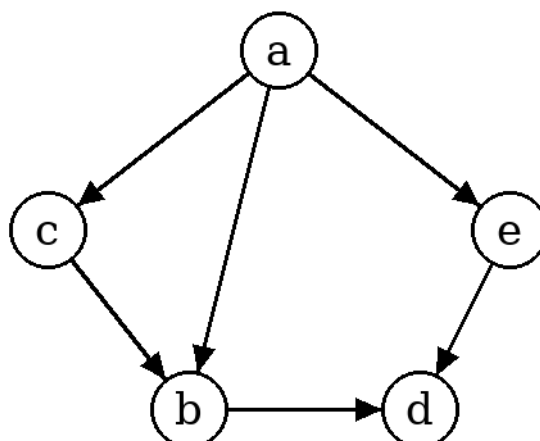
בצעו מעקב סבב אחר סבב, ורשמו את המסלול ואת אורכו (סכום משקליו).

(2) עבור גרף מכוון G ממושקל **כלשהו** שאין בו מעגל שמשקלו שלילי, הרצנו את האלגוריתם המוצא את המסלול הקצר (הקל) מקודקוד s לקודקוד t, וקיבלנו מסלול P.

- האם P יישאר בהכרח מסלול קצר ביותר בין s ל-t אם משקל **כל אחת** מן הקשתות **היוצאות מ-s יגדל** ב- x ($0 < x$)? נמקו את תשובתכם.

- האם P יישאר בהכרח מסלול קצר ביותר בין s ל-t אם משקל **כל אחת** מן הקשתות בגרף **יקטן** ב- x ($0 < x$)? נמקו את תשובתכם.

ב. (1) לפניכם גרף מכוון G :



(1) הציגו שתי סריקות DFS מקודקוד התחלה a כך שיתקבלו שני עצים שמספר העלים בהם שונה.

(2) הוסיפו מספר מינימלי של קשתות כך שבשלוש סריקות DFS מקודקוד a , יתקבלו שלושה עצים שמספרי העלים בהם שונים זה מזה. סרטטו את הגרף לאחר ההוספה, והציגו את שלושת הסריקות.

6. (25 נקודות)

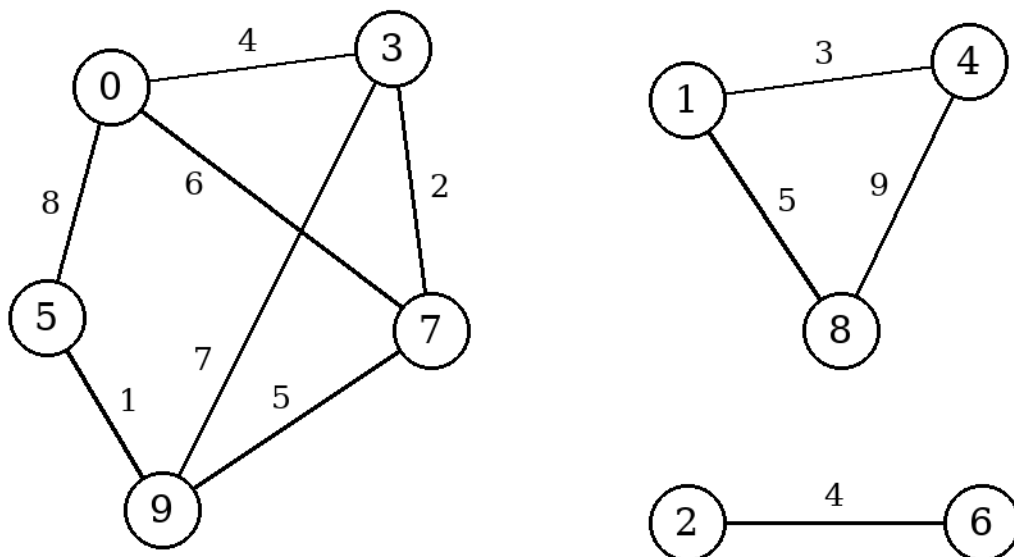
נתון גרף ממושקל, לא קשיר ולא מכוון $G(V, E)$ שבו n קודקודים, מ- v_0 עד v_{n-1} . משקלי הקשתות הם מספרים חיוביים.

הגרף מיוצג ברשימות סמיכויות: לכל קודקוד יש רשימה של שכניו, ולכל שכן - משקל הקשת אליו.

א. נתונים קודקוד v_j ומספר k . כתבו אלגוריתם המוצא ומחזיר את כל הקודקודים שיש ביניהם ובין v_j מסלול שמשקל כל אחת מן הקשתות בו הוא k לכל היותר.

הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

דוגמה: עבור הגרף שלפניכם, הקודקוד 0 ו- $k = 4$, האלגוריתם יחזיר את הקודקודים 3, 7.

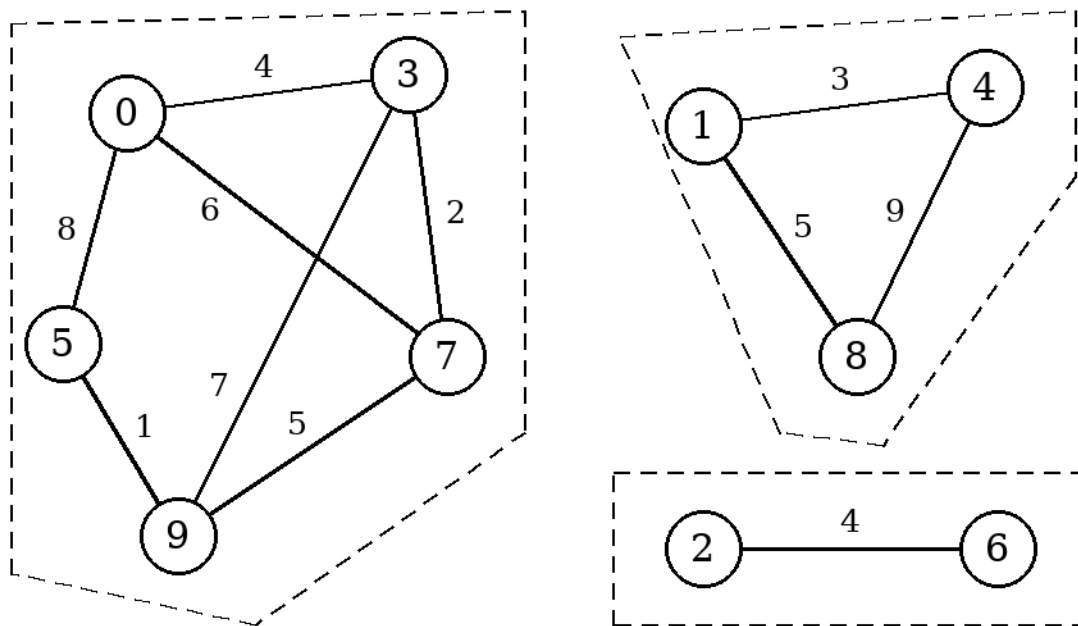


הסבר: במסלולים 0-3 ו-0-3-7 משקל כל קשת הוא 4 לכל היותר. בכל מסלול מקודקוד 0 אל אחד מן הקודקודים 5, 9 יש קשת שמשקלה גדול מ-4, ואל שאר הקודקודים אין מסלול בכלל.

ב. "רכיב קשירות" בגרף לא מכוון הוא קבוצת קודקודים שבה בין כל שני קודקודים יש מסלול, ואין שום קשת היוצאת מקודקוד בקבוצה לקודקוד שאינו בקבוצה. קודקוד שאין קשת בינו ובין שום קודקוד אחר יהיה בקבוצה משלו.

כתבו אלגוריתם המוצא את רכיב הקשירות הגדול ביותר (כלומר את הקבוצה שבה המספר המקסימלי של קודקודים) בגרף $G(V, E)$, ומחזיר את הקשתות של עץ פורש מינימלי של רכיב זה.

דוגמה: בגרף שבדוגמה לעיל יש שלושה רכיבי קשירות, והם מסומנים בקו מקווקו. הגדול ביותר הוא $\{0, 3, 5, 7, 9\}$, והאלגוריתם יחזיר את הקשתות $(0, 5)$, $(3, 7)$, $(7, 3)$, $(0, 3)$, $(9, 7)$ (משקלן הכולל 12).



הניחו שיש רק רכיב קשירות אחד שהוא הגדול ביותר.
הערה: יש לכתוב אלגוריתם יעיל שאינו עובר על כל המסלולים האפשריים.

פרק שני (50 נקודות) - מודלים חישוביים

יש לענות על שתי שאלות במסלול שלמדתם (לכל שאלה - 25 נקודות).

7. (25 נקודות)

בשאלה זו שני סעיפים, א-ב, שאין ביניהם קשר. ענו על שני הסעיפים.

א. (1) לפניכם שתי טענות (1)-(2) בנוגע לשפות שמעל הא"ב $\{a, b\}$.

בעבור כל טענה, ציינו אם היא נכונה או לא נכונה. אם הטענה נכונה - נמקו מדוע,

ואם היא לא נכונה - הביאו דוגמה נגדית. אין קשר בין הטענות.

(1) אם L_1 היא שפה רגולרית ו- L_2 היא שפה לא רגולרית, בהכרח $L_1 \cup L_2$ היא שפה לא רגולרית.

(2) לכל שפה L מתקיים: $L^* \cdot L^* = L$.

ב. (1) נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$\{w \mid w = L, \text{ הרצפים } ba, cc, \text{ אינם מופיעים ב-} w, \#b(w) + \#c(w) = \text{מספר אי-זוגי}\}$

$\#b(w)$ מציין את מספר המופעים של b במילה w .

$\#c(w)$ מציין את מספר המופעים של c במילה w .

דוגמה למילה בשפה L היא $abcb$, כי סכום המופעים של האות (2) b והאות (1) c הוא מספר

אי-זוגי (3), והרצפים ba ו- cc אינם מופיעים במילה.

(1) לפניכם חמש מילים. בנוגע לכל אחת מהן ציינו אם המילה שייכת לשפה L או לא. נמקו את

תשובתכם.

$aabcb, bba, \varepsilon, bcb, cab$.

(2) בנו אוטומט סופי דטרמיניסטי שאינו מלא המקבל את השפה L .

8. (25 נקודות)

נתונה השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{a^n b^m c^k \mid$$

$$n \bmod 2 = 0: m = n, k \bmod 2 = 0, n > 0$$

$$n \bmod 2 = 1: k = n, m \bmod 2 = 0$$

$\{$

(בשני המקרים $m \geq 0$ ו- $k \geq 0$).

דוגמה למילה בשפה L בעבור n זוגי: $aaaa bbbb cc$ ($n=4, m=4, k=2$).

הסבר: $n \bmod 2 = 0$, ובהתאם לכך מתקיים $k \bmod 2 = 0$, וגם $m = n = 4$.

דוגמה למילה בשפה L בעבור n אי-זוגי: $aaa bb ccc$ ($n=3, m=2, k=3$).

הסבר: $n \bmod 2 = 1$, ובהתאם לכך מתקיים $k = n = 3$, וגם $m \bmod 2 = 0$.

א. לפניכם חמש מילים. בעבור כל אחת מהן, ציינו אם המילה שייכת לשפה L . נמקו את תשובתכם.

$aabbc, abbbbc, aaaabbbc, aabbcacc, aaabbbcc$

ב. כתבו את המילה הקצרה ביותר בשפה L בעבור n זוגי, ואת המילה הקצרה ביותר בשפה L בעבור n אי-זוגי שיש בה לפחות אות b אחת.

ג. לפניכם אוטומט מחסנית דטרמיניסטי המקבל את השפה L . באוטומט קיימים כל המצבים,

המעברים וסימני הקלט הנדרשים. אולם מלבד המעברים בין q_0 ל- q_1 , חסרות במעברים האות

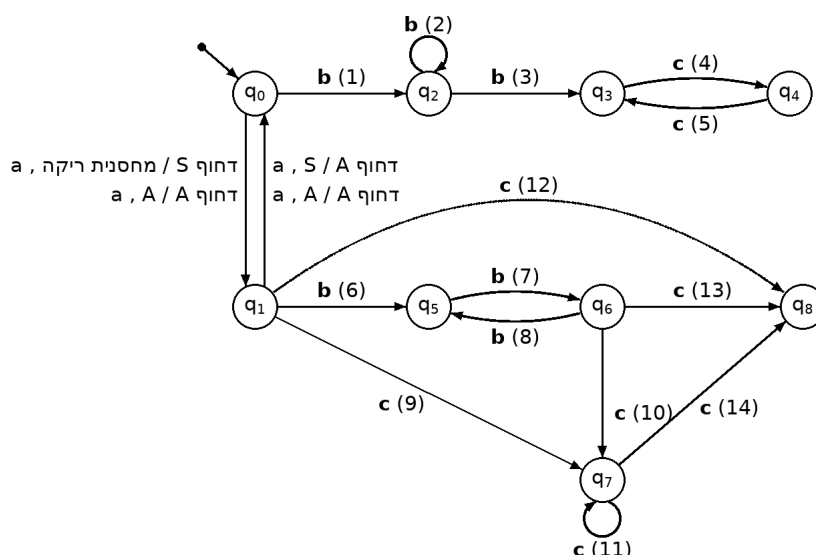
שבראש המחסנית והפעולה על המחסנית (מעברים אלה ממוספרים מ-1 עד 14). נוסף על כך,

המצבים המקבלים באוטומט לא סומנו.

בעבור כל מעבר ממוספר, ציינו את מספרו והשלימו את הסימון החסר: האות שבראש המחסנית

והפעולה שעל המחסנית (דחוף/push או שלוף/pop או ללא שינוי). נוסף על כך, כתבו מה הם

המצבים המקבלים באוטומט (אין צורך להעתיק את האוטומט למחברתכם).



9. (25 נקודות)

א. נתונה השפה L_1 מעל הא"ב $\{a, b, c\}$:

$$L_1 = \{a^n b^k a^k c^{2n} \mid n, k \geq 1\}$$

א. (1) מהי המילה הקצרה ביותר בשפה L_1 ?

(2) בנו אוטומט מחסנית דטרמיניסטי המקבל את השפה L_1 .

ב. נגדיר את השפה L_2 מעל הא"ב $\{a, b, c\}$:

$$L_2 = R(L_1) \cdot L_1$$

בעבור כל אחת מן המילים שלפניכם, ציינו אם המילה שייכת לשפה L_2 , ונמקו.

(1) ccabaabacc

(2) ccabaabaacc

(3) ccccabaaabbaacc

הערה: אין צורך לבנות אוטומט בעבור השפה L_2 .

תכנות מונחה עצמים

יש לענות על שתי שאלות במסלול שלמדתם (לכל שאלה - 25 נקודות).

10. (25 נקודות)

נוסח Java

לשאלה זו שני נוסחים: בשפת Java ובשפת C#.

לפותרים בשפת Java

במוזיאון "אמנות לכולם" פותחה מערכת ממוחשבת ובה שומרים נתונים על המוצגים. המערכת כוללת את המחלקות האלה: מוצג - **Exhibit**, ציור - **Painting**, דיוקן (ציור של אדם) - **Portrait**, פסל - **Statue**, פסל ראש - **Bust**.

נתון חלק ממבנה המחלקות במערכת הממוחשבת:

Java

```
public class Exhibit {
    private String name;
    private int year;
    public Exhibit (String name, int year)
    {
        this.name = name;
        this.year = year;
    }
    public String toString() {
        return name + ", " + year;
    }
}

public class Painting {
    private String name;
    private int year;
    private String artist;
    public void light () {
        System.out.println ("Light on " + artist);
    }
}

public class Portrait {
    private String name;
    private int year;
    private String artist;
    private String person;           // מי מצויר בדיוקן
    public void light () {
        System.out.println ("Light on " + artist);
    }
}

public class Statue {
    private String name;
    private int year;
    private String material;         // חומר: אבן/ברונזה/שיש
}

public class Bust {
    private String name;
    private int year;
    private String material;         // חומר: אבן/ברונזה/שיש
    private int height;              // גובה בסנטימטרים
}
```

הניחו שיש פעולות get ו-set לתכונות המחלקות.

א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Painting**, **Portrait**, **Statue** ו-**Bust**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה **Exhibit**). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות get ו-set קיימות.

נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ <---.

ב. נתונה המחלקה Tester :

Java

```
public class Tester {
    public static void main (String[] args) {
        Exhibit[] hall = new Exhibit [5];
        hall [0] = new Portrait ("Queen", 1720, "Ruth", "Esther");
        hall [1] = new Statue ("Rider", 1904, "bronze");
        hall [2] = new Bust ("Poet", 1860, "marble", 55);
        hall [3] = new Painting ("Garden", 1889, "Vincent");
        hall [4] = new Bust ("Sailor", 1911, "stone", 30);
        /**
    }
}
```

הוסיפו פעולות בונות למחלקות **Portrait**, **Statue**, **Painting** ו- **Bust**, המאתחלות את תכונות המחלקות (כך שהפעולה main תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של **Exhibit**.

ג. (1) בפעולה main נוסף קטע הקוד שלפניכם (במקום המסומן ב- *):

Java

```
for (int i = 0; i < hall.length; i++) {
    System.out.println (hall[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
Queen, 1720 by Ruth
Statue: Rider, 1904 [bronze]
Large bust of Poet
Garden, 1889 by Vincent
Small bust of Sailor
```

פסל ראש שגובהו 50 ס"מ לפחות נחשב גדול (Large), ופסל ראש נמוך יותר נחשב קטן (Small). הוסיפו את הפעולה toString במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה toString במחלקה **Exhibit**. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

(2) במחלקה Tester נוספה הפעולה שלפניכם:

Java

```
public static void lightAll (Exhibit [] arr) {
    for (int i = 0; i < arr.length; i++) {
        arr[i].light(); }
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך hall (הנתון לעיל), ואם הפעולה לא תקינה - תקנו אותה (כך שהקריאה ל- light בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך hall.

הערות:

. אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.

. הפעולה צריכה לרוץ ללא שגיאות עבור כל מערך מטיפוס Exhibit .

נוסח #C

לשאלה זו שני נוסחים: בשפת Java ובשפת #C .

לפותרים בשפת #C

במוזיאון "אמנות לכולם" פותחה מערכת ממוחשבת ובה שומרים נתונים על המוצגים. המערכת כוללת את המחלקות האלה: מוצג - **Exhibit**, ציור - **Painting**, דיוקן (ציור של אדם) - **Portrait**, פסל - **Statue**, פסל ראש - **Bust**. נתון חלק ממבנה המחלקות במערכת הממוחשבת:

```
C#
public class Exhibit {
    private string name;
    private int year;
    public Exhibit (string name, int year)
    {
        this.name = name;
        this.year = year;
    }
    public override string ToString() {
        return name + ", " + year;
    }
}
public class Painting {
    private string name;
    private int year;
    private string artist;
    public void Light () {
        Console.WriteLine ("Light on " + artist);
    }
}
public class Portrait {
    private string name;
    private int year;
    private string artist;
    private string person;           // מי מצויר בדיוקן
    public void Light () {
        Console.WriteLine ("Light on " + artist);
    }
}
public class Statue {
    private string name;
    private int year;
    private string material;         // חומר: אבן/ברונזה/שיש
}
public class Bust {
    private string name;
    private int year;
    private string material;         // חומר: אבן/ברונזה/שיש
    private int height;              // גובה בסנטימטרים
}
```

הניחו שיש פעולות Get ו-Set לתכונות המחלקות.

- א. כתבו מחדש את הכותרות, התכונות והפעולות של המחלקות: **Painting**, **Portrait**, **Statue** ו- **Bust**, לפי עקרונות תכנות מונחה עצמים (אין לשנות את המחלקה Exhibit). בסעיף זה אין צורך לכתוב פעולות בונות. הניחו שהפעולות Get ו- Set קיימות.
- נוסף על כך, סרטטו תרשים הייררכייה בין כל המחלקות. יש לסמן ירושה באמצעות החץ <---.
- ב. נתונה המחלקה Tester:

```
C#
public class Tester {
    public static void Main (string[] args) {
        Exhibit[] hall = new Exhibit [5];
        hall [0] = new Portrait ("Queen", 1720, "Ruth", "Esther");
        hall [1] = new Statue ("Rider", 1904, "bronze");
        hall [2] = new Bust ("Poet", 1860, "marble", 55);
        hall [3] = new Painting ("Garden", 1889, "Vincent");
        hall [4] = new Bust ("Sailor", 1911, "stone", 30);
        /**/
    }
}
```

הוסיפו פעולות בונות למחלקות **Painting**, **Portrait**, **Statue** ו- **Bust**, המאתחלות את תכונות המחלקות (כך שהפעולה Main תרוץ ללא שגיאות).

הערה: אין לשנות את הפעולה הבונה הנתונה של Exhibit.

- ג. (1) בפעולה Main נוסף קטע הקוד שלפניכם (במקום המסומן ב- *):

```
C#
for (int i = 0; i < hall.Length; i++) {
    Console.WriteLine (hall[i]); }
```

קטע הקוד תקין ומדפיס את הפלט שלפניכם:

```
Queen, 1720 by Ruth
Statue: Rider, 1904 [bronze]
Large bust of Poet
Garden, 1889 by Vincent
Small bust of Sailor
```

פסל ראש שגובהו 50 ס"מ לפחות נחשב גדול (Large), ופסל ראש נמוך יותר נחשב קטן (Small). הוסיפו את הפעולה ToString במחלקות (רק היכן שיש צורך) לפי עקרונות תכנות מונחה עצמים, כך שהפלט יהיה מתאים.

הערה: אין לשנות את הפעולה ToString במחלקה Exhibit. תשובה שבה יהיה שינוי לא תזוכה בנקודות.

- (2) במחלקה Tester נוספה הפעולה שלפניכם:

```
C#
public static void LightAll (Exhibit [] arr) {
    for (int i = 0; i < arr.Length; i++) {
        arr[i].Light(); }
}
```

אם הפעולה תקינה, כתבו את הפלט שלה עבור המערך hall (הנתון לעיל), ואם הפעולה לא תקינה - תקנו אותה (כך שהקריאה ל-Light בתוך הלולאה תהיה תקינה) וכתבו את הפלט עבור המערך hall.

הערות:

- אין לשנות את המחלקות עצמן. תשובה שבה יהיה שינוי במחלקות לא תזוכה בנקודות.
- הפעולה צריכה לרוץ ללא שגיאות עבור כל מערך מטיפוס Exhibit.

11. (25 נקודות)

נוסח Java

לשאלה זו שני נוסחים: בשפת Java ובשפת #C .

לפותרים בשפת Java

נתונות המחלקות Lamp , Led , Strip :

Java

```
public class Lamp {
    public static int total = 0;
    private int watt;
    public Lamp (int w) {
        watt = w;
        total = total + w;
    }
    public Lamp() {
        this (40);
    }
    public String toString() {
        return "[" + watt + "W/" + total + "W]";
    }
}

-----

public class Led extends Lamp {
    private String color;
    public Led (String c) {
        super (8);
        color = c;
    }
    public Led() {
        color = "white";
    }
    public String toString() {
        return color + super.toString();
    }
}

-----

public class Strip extends Led {
    private int meters;
    public Strip (int m) {
        meters = m;
    }
    public String toString() {
        return meters + "m " + super.toString();
    }
}
```

א. נתונה המחלקה Test :

Java

```
public class Test
{
    public static void main (String[] args)
    {
        Lamp[] arr = new Lamp[6];
        arr[0] = ... // (1)
        arr[1] = ... // (2)
        arr[2] = ... // (3)
        arr[3] = ... // (4)
```

```

arr[4] = ... // (5)
arr[5] = ... // (6)
for (int i = 0; i < arr.length; i++) {
    System.out.println (arr[i]);
}
}

```

השלימו את השורות (1)-(6) כך שהפלט יהיה:

```

red[8W/196W]
3m white[40W/196W]
[60W/196W]
white[40W/196W]
blue[8W/196W]
2m white[40W/196W]

```

ב. הכותרת של המחלקה Strip השתנתה לכוותרת:

Java

```
public class Strip extends Lamp
```

האם שינוי זה יגרום לשגיאה? אם כן - הסבירו את השגיאה, אם לא - רשמו מה יהיה הפלט של הפעולה main אחרי השינוי (השורות החסרות הן כפי שכתבתם בסעיף א).
הערה: אין שינויים נוספים במחלקה.

נוסח C#

לשאלה זו שני נוסחים: בשפת Java ובשפת #C.

לפותרים בשפת #C

נתונות המחלקות Lamp, Led, Strip :

```
C#
public class Lamp {
    public static int total = 0;
    private int watt;
    public Lamp (int w) {
        watt = w;
        total = total + w;
    }
    public Lamp() : this (40) {
    }
    public override string ToString() {
        return "[" + watt + "W/" + total + "W]";
    }
}

-----

public class Led : Lamp {
    private string color;
    public Led (string c) : base (8) {
        color = c;
    }
    public Led() {
        color = "white";
    }
    public override string ToString() {
        return color + base.ToString();
    }
}

-----

public class Strip : Led {
    private int meters;
    public Strip (int m) {
        meters = m;
    }
    public override string ToString() {
        return meters + "m " + base.ToString();
    }
}
}
```

א. נתונה המחלקה Test :

```
C#
public class Test
{
    public static void Main (string[] args)
    {
        Lamp[] arr = new Lamp[6];
        arr[0] = ... // (1)
        arr[1] = ... // (2)
        arr[2] = ... // (3)
        arr[3] = ... // (4)
        arr[4] = ... // (5)
        arr[5] = ... // (6)
    }
}
```

```

        for (int i = 0; i < arr.Length; i++) {
            Console.WriteLine (arr[i]);
        }
    }
}

```

השלימו את השורות (1)-(6) כך שהפלט יהיה:

```

red[8W/196W]
3m white[40W/196W]
[60W/196W]
white[40W/196W]
blue[8W/196W]
2m white[40W/196W]

```

ב. הכותרת של המחלקה Strip השתנתה לכוותרת:

C#

```
public class Strip : Lamp
```

האם שינוי זה יגרום לשגיאה? אם כן - הסבירו את השגיאה, אם לא - רשמו מה יהיה הפלט של הפעולה Main אחרי השינוי (השורות החסרות הן כפי שכתבתם בסעיף א).
הערה: אין שינויים נוספים במחלקה.

12. (25 נקודות)

נוסח Java

Java

```

public class Dot {
    private static int count = 0;
    private int v;

    public Dot() { this(5); }
    public Dot(int v) {
        this.v = v;
        count++;
    }
    public int getV() { return this.v; }
    public static int getCount() { return count; }
    public void up() { this.v = this.v + 2; }
    public void step() { up(); up(); }
    public void step(int k) { this.v = this.v * k; }
    public String toString() { return "v = " + this.v; }
}

public class Ring extends Dot {
    private int r;

    public Ring() {
        super();
        this.r = 1;
    }
    public Ring(int v) {
        super(v + 1);
        this.r = v;
    }
    public Ring(Ring other) {
        super(other.getV());
        this.r = other.r + 10;
    }
    public void up() { this.r = this.r + getV(); }
    public void step(int k) {
        super.step(k);
        this.r = this.r - 1;
    }
    public String toString() {
        return "(" + super.toString() + ", r = " + this.r + ")";
    }
}

```

א. לפניכם קטע קוד:

Java

```

Dot[] arr = new Dot[5];
arr[0] = new Dot();
arr[1] = new Ring();
arr[2] = new Ring(3);
arr[3] = arr[1];
arr[4] = new Ring((Ring) arr[2]);
for (int i = 0; i < arr.length; i++)
    System.out.println(arr[i]);
System.out.println("count = " + Dot.getCount());

```

ציירו את העצמים שנוצרו, וכתבו מה הקוד מדפיס.

ב. לפניכם המשך קטע הקוד:

Java

```
arr[0].step();
arr[1].step();
arr[2].step(2);
((Dot) arr[4]).up();
Ring q = (Ring) arr[3];
q.step(3);
Dot e = new Ring(2);
e.step();
System.out.println("e: " + e);
arr[0] = e;
for (int i = 0; i < arr.length; i++)
    System.out.println(arr[i]);
System.out.println("count = " + Dot.getCount());
```

כתבו מה הקוד מדפיס.

נוסח #C

נתונות המחלקות Ring, Dot :

C#

```

public class Dot {
    private static int count = 0;
    private int v;

    public Dot() : this(5) { }
    public Dot(int v) {
        this.v = v;
        count++;
    }
    public int GetV() { return this.v; }
    public static int GetCount() { return count; }
    public virtual void Up() { this.v = this.v + 2; }
    public void Step() { Up(); Up(); }
    public virtual void Step(int k) { this.v = this.v * k; }
    public override string ToString() { return "v = " + this.v; }
}

public class Ring : Dot {
    private int r;

    public Ring() : base() {
        this.r = 1;
    }
    public Ring(int v) : base(v + 1) {
        this.r = v;
    }
    public Ring(Ring other) : base(other.GetV()) {
        this.r = other.r + 10;
    }
    public override void Up() { this.r = this.r + GetV(); }
    public override void Step(int k) {
        base.Step(k);
        this.r = this.r - 1;
    }
    public override string ToString() {
        return "(" + base.ToString() + ", r = " + this.r + ")";
    }
}

```

א. לפניכם קטע קוד:

C#

```

Dot[] arr = new Dot[5];
arr[0] = new Dot();
arr[1] = new Ring();
arr[2] = new Ring(3);
arr[3] = arr[1];
arr[4] = new Ring((Ring) arr[2]);
for (int i = 0; i < arr.Length; i++)
    Console.WriteLine(arr[i]);
Console.WriteLine("count = " + Dot.GetCount());

```

ציירו את העצמים שנוצרו, וכתבו מה הקוד מדפיס.

ב. לפניכם המשך קטע הקוד:

C#

```
arr[0].Step();
arr[1].Step();
arr[2].Step(2);
((Dot) arr[4]).Up();
Ring q = (Ring) arr[3];
q.Step(3);
Dot e = new Ring(2);
e.Step();
Console.WriteLine("e: " + e);
arr[0] = e;
for (int i = 0; i < arr.Length; i++)
    Console.WriteLine(arr[i]);
Console.WriteLine("count = " + Dot.GetCount());
```

כתבו מה הקוד מדפיס.

כל הבגרויות במדעי המחשב עם פתרונות מלאים: www.guyai.co.il/bagrut/computer-science