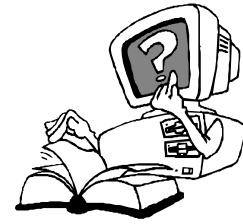


מדעי המחשב ב'

2 יח"ל (השלמה ל - 5 יח"ל)

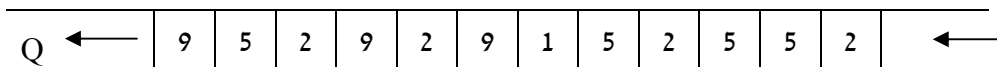


פרק ראשון (50 נקודות)

שאלה 1:

נתון תור Q שכל אבריו הם מספרים שלמים. מספר יכול להופיע בתור יותר מפעם אחת.

דוגמא:



נתונה הפעולה הבאה הפועלת על טיפוס הנתונים איבר-מרכיב.

טיפוס נתונים (טני"מ) זה כולל שני שדות מספריים: שדה ערך ושדה כמות.

אתחל-איבר-מרכיב (ערך, כמות) - פעולה המקבלת שני מספרים שלמים: ערך ו-כמות ומאתחלת משתנה מסוג איבר-מרכיב כך ששדה הערך יהיה ערך ושדה הכמות יהיה כמות.
(שים ♥ : הפעולה קיימת. ויש להשתמש בה מבלי לממש אותה).

פתח אלגוריתם המקבל את התור Q ויוצר רשימה-מרוכזת שכל איבר בה הוא מטיפוס איבר-מרכיב, כך ששדה ערך באיבר המרכיב יהיה ערך האיבר בתור, ושדה כמות באיבר יהיה מספר המופעים של איבר זה בתור. (שים ♥ : איבר-מרכיב הינו ה- list_info_type של רשימה-מרוכזת)

עבור התור שבדוגמא תוחר הרשימה הבאה:



רעיון הפתרון:

נתייחס לתור כאל "תור מעגלי".

נשלוף את האיבר שבראש התור, ועבור כל האיברים שאחריו, אם הם שווים לו הם ייספרו, אחרת הם יוחזרו לתור.

כדי לדעת מתי סיימנו מעבר על התור, נדחוף בתחילת המעבר איבר דמה לסוף התור.

בסיום הפעולה יתקבל תור ריק. אם רוצים לשמור על התור, נעבוד על עותק שלו.

סיבוכיות הפעולה:

מבחינת מקום $O(1)$ - עובדים על אותו שטח זיכרון.

מבחינת זמן ריצה: $O(n^2)$ - עבור כל איבר שטרם טופל עוברים על כל התור.

רשימה-מרכזת (Q)

- { פעולה המקבלת תור Q, ומחזירה רשימה שכל איבר בה מטיפוס איבר-מרכז המכיל ערך מהתור ומספר מופעיו.
הנחה: התור Q מאותחל.
- (1) אתחל-רשימה $L \leftarrow$
- (2) עוגן-רשימה $p \leftarrow (L)$
- (3) הכנס-לתור (Q, d) { איבר דמה = d }
- (4) הוצא-מתור (Q) $x \leftarrow$
- (5) $counter \leftarrow 1$
- (6) כל עוד לא תור-ריק? (Q) , בצע:
- (6.1) הוצא-מתור (Q) $y \leftarrow$
- (6.2) אם $y \neq d$ אזי -
- (6.2.1) אם $y = x$ אזי - $counter++$
- (6.2.2) אחרת - הכנס-לתור (Q, y)
- (6.3) אחרת -
- (6.3.1) אתחל-איבר-מרכז $(x, counter)$ $element \leftarrow$
- (6.3.2) הכנס-לרשימה $(L, p, element)$
- (6.3.3) עוקב-ברשימה (L, p) $p \leftarrow$
- (6.3.4) הכנס-לתור (Q, y) { הוכנס איבר הדמה למעבר נוסף על התור }
- (6.3.5) הוצא-מתור (Q) $x \leftarrow$
- (6.3.6) $counter \leftarrow 1$
- (7) החזר את L

הערה: ניתן לפשט את האלגוריתם על ידי כתיבת פעולת עזר: **ספור-מופעים** (Q, x) המוציאה מתור Q את כל המופעים של x ומחזירה את כמה מופעים היו. הפעולה הראשית תוציא איבר מהתור, תפעיל את הפעולה **ספור-מופעים**, תאתחל איבר-מרכז, ותכניס אותו לרשימה, עד שהתור יתרוקן.

שאלה 2:

(מקור השאלה: **רקורסיה**, חיים אברבוך ואיתן ראט, למידה בהנאה)

נתונה הפונקציה הרקורסיבית הבאה $distance$, המקבלת כפרמטר שני מספרים שלמים וחיוביים ומחזירה את הפרש מספר הספרות שביניהם.

לדוגמא: $distance(248, 24646468)$ יחזיר 5
ואילו $distance(2345, 9876)$ יחזיר 0

```
int distance (int num1, int num2)
{
    ...
}
```

ממש בסביבת העבודה את הפונקציה הרקורסיבית $distance$, מבלי להשתמש בפונקציות עזר!

רעיון הפתרון:

אם אחד המספרים שווה ל-0 אז הפרש הספרות הוא מספר הספרות של המספר השונה מ-0. לכן, כל עוד שני המספרים שונים (גדולים) מ-0, נוריד במקביל ספרה אחת מכל אחד מהמספרים, עד שאחד המספרים יתאפס. עתה, נמשיך בתהליך הורדת ספרה תוך ספירת מספר ההורדות.

הפרש-הספרות (num1, num2)

(1) אם $num1 = 0$ וגם $num2 = 0$ החזר 0

(2) אם $num1 \neq 0$ וגם $num2 \neq 0$

החזר הפרש-הספרות (num2 div 10 , num1 div 10)

(3) החזר 1 + הפרש-הספרות (num2 div 10 , num1 div 10)

הסבר: $num \div 10$ מחזיר את המספר ללא ספרת האחדות שלו (חילוק בשלמים).

הוראה (2) קריאה רקורסיבית המורידה מכל אחד מהמספרים ספרה אחת, כך שהפרש הספרות ביניהם נשמר, ושולחת כפרמטר את המספרים "המוקטנים".

הוראה (3) מבצעת את הספירה. הקריאה הרקורסיבית מורידה ספרה מהמספר השונה מ-0 ואינה מבצעת שינוי במספר השני (0 חלקי 0 נשאר 0) עבור כל קריאה רקורסיבית מתווסף 1 להפרש הספרות.

שפת C:

```

/* טענת כניסה: num1 ו-num2 מספרים שלמים וחיוביים. */
/* טענת יציאה: הפרש מספר הספרות של num1 ו-num2 */

int distance (int num1, int num2)
{
    if (num1 == 0 && num2 == 0)
        return 0;
    if (num1 != 0 && num2 != 0)
        return distance (num1 / 10, num2 / 10);
    return 1 + distance (num1 / 10, num2 / 10);
}

```

שפת פסקל:

```

/* טענת כניסה: num1 ו-num2 מספרים שלמים וחיוביים. */
/* טענת יציאה: הפרש מספר הספרות של num1 ו-num2 */

function distance (num1, num2:integer):integer;
begin
    if (num1 = 0) and (num2 = 0) then
        distance := 0
    else
        if (num1 <> 0) and (num2 <> 0) then
            distance := distance (num1 div 10, num2 div 10)
        else
            distance := 1 + distance (num1 div 10, num2 div 10);
    end;
end;

```

שאלה 3:

נתון האלגוריתם הבא הפועל על עץ בינארי:

מה-אני (T, m)

(1) אם לא עץ-ריק? (T) , אזי:

(1.1) אם עץ-ריק? (תת-עץ-שמאלי (T)) וגם עץ-ריק? (תת-עץ-ימני (T)),

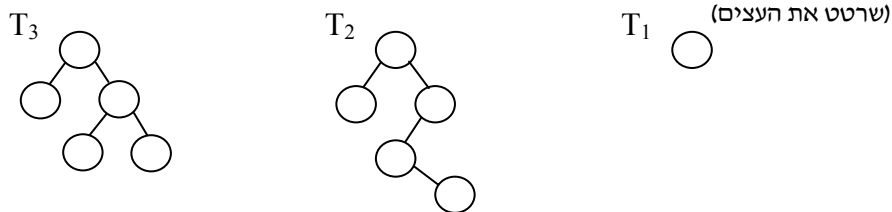
אזי: **עדכן-שורש** (T, m)

(1.2) אחרת:

(1.2.1) **מה-אני** $(m+1, \text{תת-עץ-שמאלי } (T))$

(1.2.2) **מה-אני** $(m+1, \text{תת-עץ-ימני } (T))$

א. מה תהיה תוצאת הקריאה: **מה-אני** $(T, 0)$ על העצים T_1, T_2 ו- T_3 שלהלן?



ב. הסבר מה מטרת האלגוריתם **מה-אני** (T, m)

(T) הינו עץ בינארי, ו- m מספר טבעי.

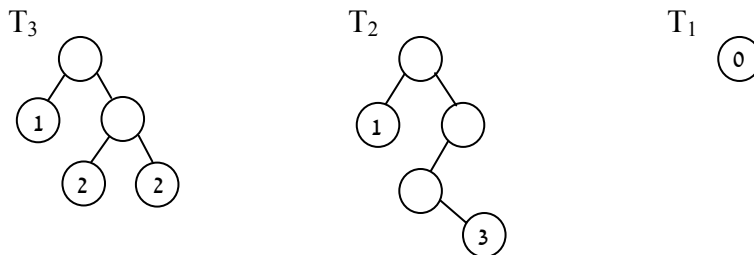
ג. יישם אלגוריתם **אחר**, בסביבת העבודה, המוצא את ערך m הגבוה ביותר בעץ **לאחר**

שהופעה עליו הפעולה **מה-אני** $(T, 0)$

שים ♥: אין לתרגם אלגוריתם זה, אלא יש ליישם פעולה **אחרת** המשיגה את אותה

המטרה שהגדרת בסעיף ב.

א. העצים לאחר הפעלת **מה-אני** $(T, 0)$



ב. האלגוריתם מעדכן בכל עלה בעץ את מספר הרמה שלו $+ m$.

ג. פעולה שקולה הינה הפעולה: **רמה-מקסימאלית** (T) או **גובה-העץ** (T)

רמה-מקסימאלית (T)

} הפעולה מקבלת עץ בינארי T ומחזירה את מספרה של הרמה המקסימאלית בעץ.

{ הנחה: T מאותחל

(1) אם עץ-ריק? (T) החזר -1

(2) אם עלה? (T) החזר 0

(3) החזר $+ 1$ מקסימום (**רמה-מקסימאלית** (תע"ש) (T)), **רמה-מקסימאלית** (תע"י) (T))

יש לממש גם את הפעולות עלה? ו-**מקסימום**.

שפת C :

```

/* הפעולה מקבלת עץ בינארי T ומחזירה את מספרה של הרמה המקסימאלי בעץ. */
/* הנחה: T מאותחל */
int max_level (tree_type T)
{
    if (tree_is_empty (T)) return -1 ;
    if (leaf (T)) return 0;
    return 1 + max (max_level (tree_Lsub (T)), max_level (tree_Rsum (T)));
}

/* הפעולה המחזירה "אמת" אם T עלה, ו-"שקר" אחרת. */
/* הנחה: T מאותחל */
/* TRUE ו- FALSE הם קבועים שערכם 1 (אמת) ו- 0 (שקר) בהתאמה */
int leaf (tree_type T)
{
    if (tree_is_empty (T)) return FALSE;
    if ( tree_is_empty (tree_Lsub(T)) && tree_is_empty (tree_Lsub(T)) ) return TRUE;
    return FALSE ;
}

/* פעולה המקבלת שני מספרים המחזירה את הגדול מביניהם. */
int max (int x, int y)
{
    if (x > y) return x;
    return y ;
}

```

שפת פסקל:

```

(* הפעולה מקבלת עץ בינארי T ומחזירה את מספרה של הרמה המקסימאלי בעץ. *)
(* הנחה: T מאותחל *)
function max_level (tree_type T):integer;
begin
    if (tree_is_empty (T)) then max_level := -1
    else if (leaf (T)) then max_level := 0
    else max_level := 1 + max (max_level (tree_Lsub (T)), max_level (tree_Rsum (T)));
end;

(* הפעולה המחזירה "אמת" אם T עלה, ו-"שקר" אחרת. *)
(* הנחה: T מאותחל *)
function leaf (T : tree_type) : boolean;
begin
    if (tree_is_empty (T)) leaf := false
    else if (tree_is_empty (tree_Lsub(T)) and (tree_is_empty (tree_Lsub(T))) leaf := true
    else leaf := false ;
end;

```

) פעולה המקבלת שני מספרים המחזירה את הגדול מביניהם. (
function **max** (x, y : integer) : integer;
begin
 if (x > y) then max := x
 else max := y ;
end;

שאלה 4:

(מקור השאלה: הכנה לבחינות בגרות 5 יח"ל, נוע רגוניס, מבט לחלונות).

הכתב לעניינים פוליטיים של העיתון "יום-טוב" החליט ליעל את עבודתו בנושא קשריו עם גורמים פוליטיים שונים. מטרת הכתב היא להגיע למקורות מידע, מבין הפוליטיקאים המוכנים לשתף פעולה, כדי שישפכו לו חומר עדכני ומעניין לכתבותיו.

בעבר היה פונה למקורות באופן אקראי, דבר שפגע בעבודתו. כיום, עקב הניסיון הרב שצבר, הוא מכיר מספר פוליטיקאים אך מידת שיתוף הפעולה של כל אחד מהם שונה.

הכתב החליט לרכז את שמות הפוליטיקאים עימם הוא בקשר ולתת ציון לכל אחד מהם. הציון הגבוה ביותר יינתן לפוליטיקאי ממנו יש את הסיכוי הגבוה ביותר להגיע למידע.

עבודת הכתב היא דינאמית, כך שמידי פעם יש צורך לעדכן את הציון של פוליטיקאי מסויים, להוסיף פוליטיקאי חדש לרשימותיו או למחוק פוליטיקאי מרשימותיו.

יתכן ומספר פוליטיקאים יקבלו אותו ציון.

- הצע דרך לייצג את טיפוס הנתונים המופשט **פוליטיקאי**.
- כתוב ממשק עברי לטיפוס הנתונים המופשט **פוליטיקאי**. הממשק יכלול את הגדרת שם הפעולות, הפרמטרים שהן מקבלות, מה הן מחזירות ותאור פעולתן.
- הצע דרך לייצג את טיפוס הנתונים **מאגר-פוליטיקאים** כך שסיבוכיות הפעולה של קבלת שם הפוליטיקאי בעל הציון הטוב ביותר תהיה $O(1)$.
- כתוב ממשק עברי לטיפוס הנתונים המופשט **מאגר-פוליטיקאים**. הממשק יכלול את הגדרת שם הפעולות, הפרמטרים שהן מקבלות, מה הן מחזירות ותאור פעולתן.
- כתוב אלגוריתם המממש פעולה המקבלת שם וציון של פוליטיקאי חדש ומכניסה אותו למאגר. ניתן להניח שהפוליטיקאי עדיין לא נמצא במאגר.

פוליטיקאי = מבנה / רשומה
שם = מחרוזת
ציון = מספר

פעולה המחזירה פוליטיקאי מאותחל אשר שמו שם, וציונו ציון.	אתחל-פוליטיקאי (שם, ציון)
פעולה המחזירה את שם הפוליטיקאי. הנחה: פוליטיקאי מאותחל.	אחזר-שם (פוליטיקאי)
פעולה המחזירה את ציונו של פוליטיקאי. הנחה: פוליטיקאי מאותחל.	אחזר-ציון (פוליטיקאי)
פעולה המעדכנת את ציונו של הפוליטיקאי לציון החדש. הנחה: פוליטיקאי מאותחל	עדכן-ציון (ציון, פוליטיקאי)

מאגר-פוליטיקאים = רשימה ממויינת שכל אחד מאיבריה הוא מטיפוס פוליטיקאי.
הערה: ניתן לייצג את המאגר בתור או המחסנית ממויינים.

אתחל-מאגר	פעולה המחזירה מאגר ריק.
מאגר-ריק? (מאגר)	פעולה המחזירה "אמת" אם המאגר ריק, ו-"שקר" אחרת. הנחה: המאגר מאותחל.
פוליטיקאי-בעל-ציון-גבוה (מאגר)	פעולה המחזירה את הפוליטיקאי בעל הציון הגבוה ביותר. הנחה: המאגר מאותחל ואינו ריק.
רשימת-פוליטיקאים-בעלי-ציון (ציון, מאגר)	פעולה המחזירה רשימה של כל הפוליטיקאים במאגר שציונם ציון. אם אין כאלו פוליטיקאים תוחזר רשימה ריקה. הנחה: המאגר מאותחל.
הוסף-פוליטיקאי-למאגר (שם, ציון, מאגר)	פעולה המקבלת שם וציון של פוליטיקאי ומוסיפה אותו למאגר במקומו הנכון לפי הציון. הנחה: המאגר מאותחל.
פוליטיקאי-במאגר? (שם, מאגר)	פעולה המחזירה "אמת" אם הפוליטיקאי אשר שמו שם נמצא במאגר ו-"שקר" אחרת. הנחה: המאגר מאותחל.
מחק-פוליטיקאי (שם, מאגר)	פעולה המקבלת שם של פוליטיקאי ומסירה אותו מהמאגר אם הוא נמצא בו. הנחה: המאגר מאותחל. הפוליטיקאי נמצא במאגר.
עדכן-פוליטיקאי-במאגר (שם, ציון, מאגר)	פעולה המקבלת שם של פוליטיקאי מעדכנת את ציונו לציון, ומעבירה אותו למקום חדש לפי הציון החדש. הנחה: המאגר מאותחל, הפוליטיקאי נמצא במאגר.

הוסף-פוליטיקאי-למאגר (שם, ציון, מאגר)

{ פעולה המקבלת שם וציון של פוליטיקאי ומוסיפה אותו למאגר במקומו הנכון לפי הציון.
הנחה: המאגר מאותחל. }

(1) אתחל-פוליטיקאי (שם, ציון) ← פוליטיקאי

(2) "שקר" ← עודכן

(3) עוגן-רשימה ← p

(4) כל עוד עוקב-ברשימה (p, מאגר) ≠ סוף-רשימה (מאגר) וגם לא עודכן, בצע:

(4.1) אחזר-מרשימה (עוקב-ברשימה (מאגר), מאגר) ← x

(4.2) אם אחזר-ציון (x) גדול מציון אזי,

(4.2.1) עוקב-ברשימה (p, מאגר) ← p

(4.3) אחרת -

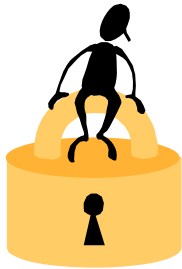
(4.3.1) הכנס-לרשימה (פוליטיקאי, p, מאגר)

(4.3.2) "אמת" ← עודכן

(5) אם לא עודכן, אזי הכנס-לרשימה (פוליטיקאי, p, מאגר) { למקרה והמאגר היה ריק }

במימוש מחסנית יש לשלוף את כל האיברים הגדולים מהציון למחסנית עזר ואחר כך להחזירם למחסנית.
במימוש תור יש להעתיק את כל האיברים לתור עזר (עד שיתרוקן) ואחר כך להחזירם לתור.

אם מבצעים הרבה פעולות הכנסה והוצאה, יהיה המימוש היעיל ביותר מימוש מחסנית (בגלל העלות הגבוהה של הכנסה והוצאה מרשימה) ואם מבצעים הרבה חיפושים, המימוש היעיל ביותר הוא של רשימה.



פרק שני (50 נקודות)

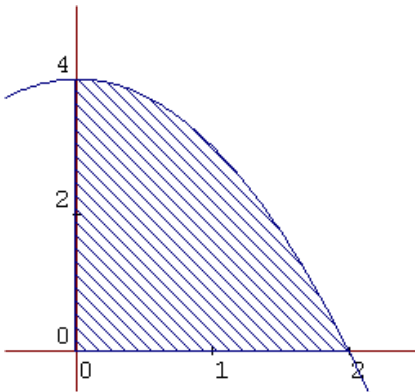
בפרק זה שאלות משישה מסלולים שונים. עליך לענות רק על השאלות במסלול שלמדת, על-פי ההוראות באותו המסלול.

תורת המחשב

ענה על שתיים מהשאלות 9 - 12 (לכל שאלה 25 - נקודות).

שאלה 9:

נתון גרף הפונקציה הבאה: $y = -x^2 + 4$ בתחום $[0, 2]$.



- חשב לכמה טרפזים יש לחלק את השטח על מנת לקבל פתרון מקורב לשטח המסומן בדיוק של 0.0001?
- חשב (ידינית) את השטח הכלוא בין הגרפים של שתי הפונקציות בתחום $[0, 2]$ (מקווקו באיור) לפי שיטת הטרפזים, עבור $n = 4$.
- מהי השגיאה המוחלטת ומהי השגיאה היחסית של התוצאה שקיבלת בסעיף ב, אם השטח המדויק הוא $5\frac{1}{3}$?
- מספר הטרפזים הדרוש בשגיאה המותרת חסום על ידי הביטוי:

$$n \geq \sqrt{\frac{|\max[a, b]F''(x)| \cdot (b - a)^3}{12 \cdot Err}}$$

$$|\max[a, b]F''(x)| = |(-x^2 + 4)''| = |-2| = 2$$

$$n \geq \sqrt{\frac{2 \cdot (2 - 0)^3}{12 \cdot 0.0001}} = 115.47 \Rightarrow n = 116$$

ב. $n = 4$, $[0, 2]$
 $w = \frac{b - a}{n} = \frac{2 - 0}{4} = 0.5$

$$S = \frac{w}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(a + iw) + f(b) \right]$$

$$S = \frac{0.5}{2} [f(0) + 2 \cdot (f(0.5) + f(1) + f(1.5)) + f(2)]$$

הקירוב לשטח הוא $5\frac{1}{4}$.

$$\frac{1}{4} [4 + 2 \cdot (3.75 + 3 + 1.75) + 0] = 5\frac{1}{4}$$

ג. שגיאה מוחלטת: $\left| 5\frac{1}{3} - 5\frac{1}{4} \right| = \frac{1}{12} = 0.08333$

$$\frac{0.08333}{5\frac{1}{3}} * 100\% = 1.5625\% \quad \text{שגיאה יחסית:}$$

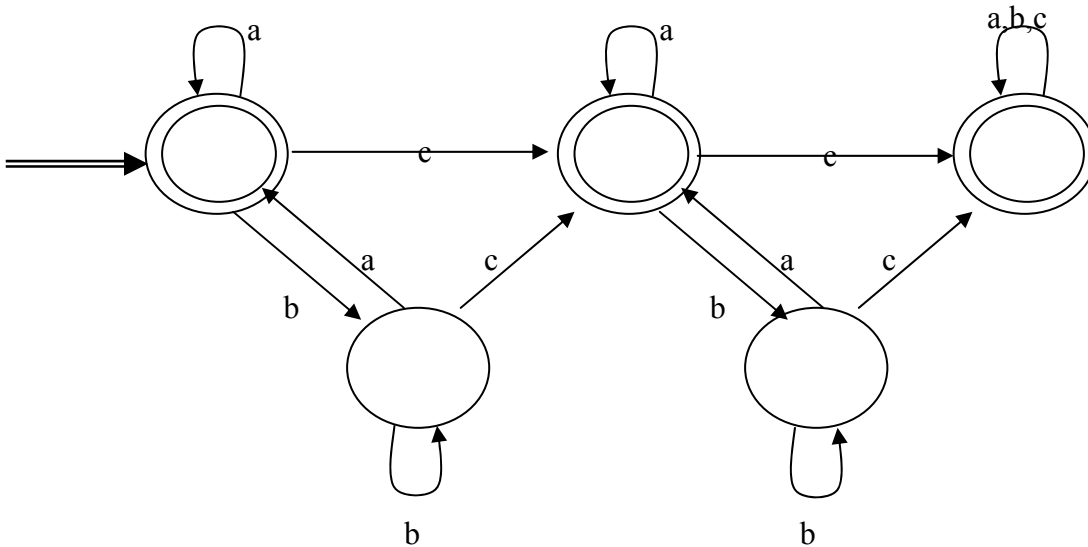
שאלה 10:

שאלה בתורת הגרפים ...

שאלה 11:

בנה אוטומט סופי דטרמיניסטי לשפה הבאה:

תהי L השפה מעל $\Sigma = \{a, b, c\}$ המכילה את כל המילים כך שאם הן מסתיימות ב- b אזי הן מכילות לפחות שני מופעים של c . (שים לב: מילה בשפה עשויה שלא להסתיים ב- b).



שאלה 12:

א. לפינת השפה L מעל א"ב $\{a, b, c\}$:

$$L = \{ a^m b^n c^k \mid m > 0, n \geq m, k = n - m, \text{ מספר ה- } k \text{ מתחלק ב-} 3 \}$$

בנה אוטומט מחסנית שיקבל את השפה L .

ב. כתוב דקדוק עבור השפה L .

אם: $k = n - m$ אזי: $n = k + m$. נכתוב שפה שקולה:

$$L = \{ a^m b^{m+k} c^k \mid m > 0, \text{ מספר ה- } k \text{ מתחלק ב-} 3 \}$$

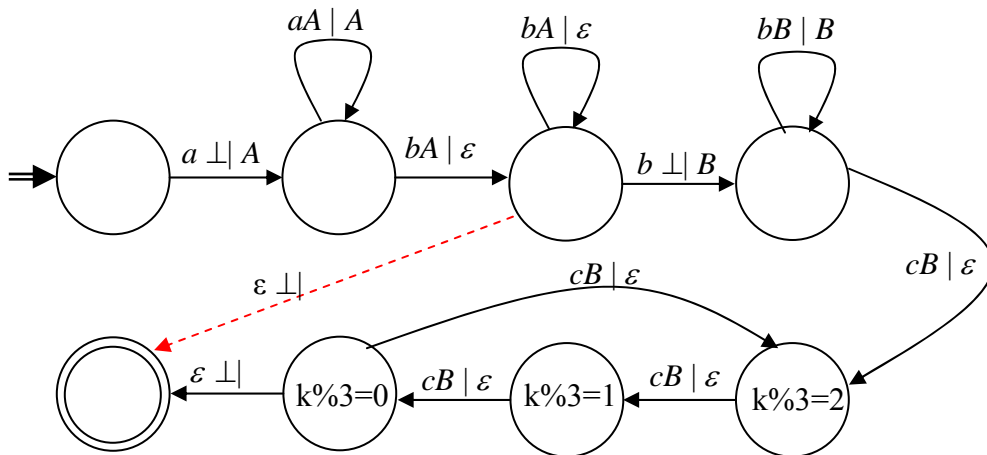
$$L = \{ a^m b^m b^k c^k \mid m > 0, \text{ מספר ה- } k \text{ מתחלק ב-} 3 \}$$

עבור כל a נדחוף A למחסנית.

עבור כל b נשלוף A מהמחסנית, עד שהמחסנית תתרוקן.

לאחר מכן, עבור כל b נדחוף B למחסנית.

עבור כל c נשלוף B אך נספור את מספר ההוצאות כך שיתחלק ב-3. (שים לב: k יכול להיות גם 0).



$k \% 3 = 2$ שארית החלוקה של k ב-3 היא 2. ($\text{mod} = \%$)

במחסנית יש k סימני B. על פי הדרישה, מספר ה-B צריך להתחלק ב-3 ללא שארית. כל שליפה של B מהמחסנית, משנה את השארית.

נציג דקדוק לשפה:

$$L = \{ a^m b^m b^k c^k \mid m > 0, n \geq m, k \text{ מתחלק ב-} 3 \}$$

G: $S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow bCc \mid \epsilon$

$C \rightarrow bDc$

$D \rightarrow bBc$

$B \rightarrow bbbBccc \mid \epsilon$ אפשר באחת משתי הצורות:

הערה: אם במקום $n \geq m$ יהיה $n > m$:

השינויים באוטומט המחסנית: נסיר את המעבר המקווקו (מ- q_2 למצב מסיים)

השינויים בדקדוק: $B \rightarrow bbbBccc \mid bbbccc$

בהצלחה!
היזה

