

# מבחן בגרות 2006

## פרטים כלליים

מועד הבחינה: בכל זמן

מספר השאלון: 1

משך הבחינה: 3 שעות

חומר עזר בשימוש: הכל (ספרים ומחברות)

המלצות: קרא המלצות לפני הבחינה ובדיקות אחרונות לפני מסירה (עמודים 8-11)

## מבנה השאלון

סה"כ 50 נקודות	2 שאלות (בחירה מ-4) לכל שאלה – 25 נקודות	פרק ראשון - עיצוב תוכנה
סה"כ 50 נקודות	2 שאלות (בחירה מ-4) לכל שאלה – 25 נקודות	פרק שני - מודלים חישוביים

## תוכן עניינים של פתרון המבחן

### פרק ראשון - עיצוב תוכנה

שאלה 1: תור, סיבוכיות [פיתוח פעולה] .....

שאלה 2: מחסנית [מעקב] .....

שאלה 3: רקורסיה [פיתוח פעולה] .....

שאלה 4: רשימה [פיתוח פעולות] .....

### פרק שני - מודלים חישוביים

שאלה 13: אוטומט מחסנית [בניית מודל] .....

שאלה 14: פעולות על מילים ושפות [זיהוי מילים / קביעת רגולריות] .....

שאלה 15: פעולות על מילים ושפות [שייכות מילים לשפה, הרכבת שפות, קביעת נכונות של טענות]

שאלה 16: אוטומט סופי דטרמיניסטי [בניית מודל] .....

כתוב/ סרק ראשון - ע'צ'אב תאכנה 2006

פתרון שאלה 1

נושא מרכזי: תור  
סוג השאלה: פיתוח פעולה

א. אלגוריתם שממש את הפעולה תור לפי שכיחות (Q)  
נשתמש בשתי פעולות עזר:

▪ פעולה המעתיקה תור לתור חדש

```
public static Queue<int> CloneQueue(Queue<int> q)
{ // הפעולה מקבלת תור ומחזירה תור חדש זהה לו
  Queue<int> clone = new Queue<int>();
  Queue<int> help = new Queue<int>();
  int num;
  while (!q.IsEmpty())
  {
    num = q.Remove();
    clone.Insert(num);
    help.Insert(num);
  }
  while (!help.IsEmpty())
  {
    q.Insert(help.Remove());
  }
  return clone;
}
```

▪ פעולה הבודקת כמה איברים בתור שווים לערך שהתקבל כפרמטר, ומוציאה אותם מהתור.

```
public static int CountInQueue(Queue<int> q, int x)
{ // הפעולה מקבלת תור של מספרים שלמים, ומספר המופיע בתור
  // הפעולה מחזירה את השכיחות של המספר בתור ומוחקת את המספר וכל מופעיו
  Queue<int> help = new Queue<int>();
  int y, count = 0;
  while (!q.IsEmpty())
  {
    y = q.Remove(); // הוצאת איבר מהתור
    if (x == y)
      count++; // במקרה של שוויון, הגדלת המונה ב-1
    else
      help.Insert(y); // במקרה שהאיבר הנוכחי שינו שווה, יש להעבירו לתור העזר
  }
```

```

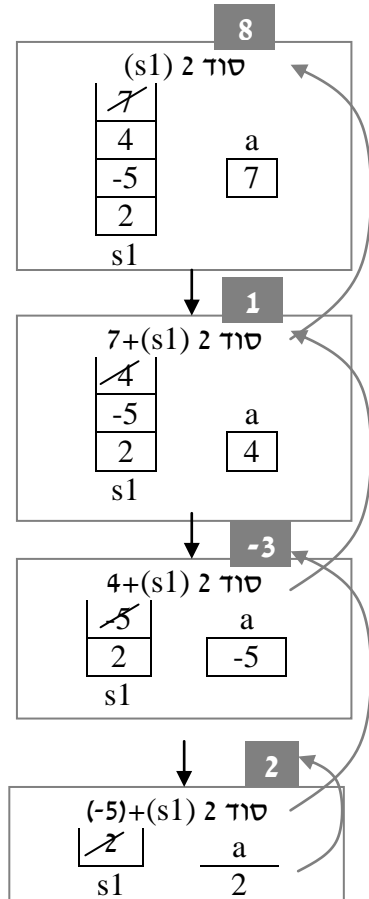
}
while (!help.IsEmpty())
{
    q.Insert(help.Remove()); // החזרת כל האיברים השונים מ-x לתור המקורי
}
return count; // הפעולה מחזירה את מספר האיברים בתור שהיו שווים ל-x והוצאו מהתור
}
public static Queue<int> QueueByFreq(Queue<int> q)
{ // הפעולה מקבלת תור של מספרים שלמים ומחזירה תור לפי שכיחות
    q = CloneQueue(q); // הפניה להעתק של התור כדי לשמור על המקורי
    Queue<int> freq = new Queue<int>(); // תור לפי שכיחות
    int num;
    while (!q.IsEmpty())
    {
        num = q.Head();
        freq.Insert(num); // הכנסת המספר לתור השכיחות
        freq.Insert(CountInQueue(q, num)); // הכנסת שכיחות המספר לתור השכיחות
    }
    return freq;
}

```

- ב. הסיבוכיות היא  $O(n^2)$  כאשר  $n$  מייצג את מספר האיברים בתור. מתבצעת סריקה של כל אברי התור ועבור כל איבר מתבצעת סריקה נוספת כדי למנות את מספר המופעים שלו בתור. סיבוכיות כל פעולות התור היא  $O(1)$ .

## פתרון שאלה 2

נושא מרכזי: מחסנית  
סוג השאלה: מעקב

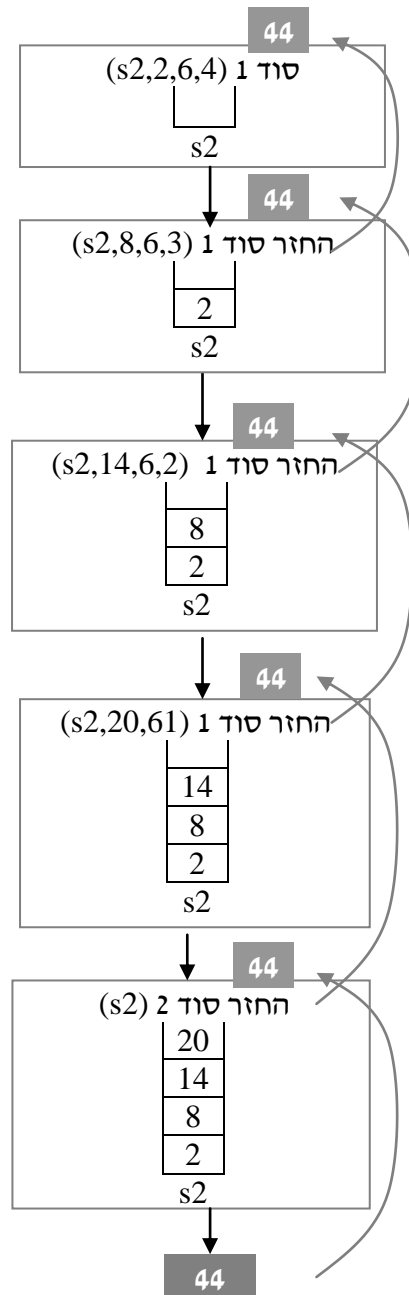


ב

האלגוריתם sod2 מחזיר את סכום האיברים במחרוזת.

ג

האלגוריתם מחזיר 44.



ד

האלגוריתם סוד1 מחזיר את סכום n האיברים הראשונים של סדרה שהאיבר הראשון שלה הוא a וההפרש d



### פתרון שאלה 3

נושא מרכזי: רקורסיה  
סוג השאלה: פיתוח פעולה

#### תת תכנית רקורסיבית Squares

```
public static void Squares(int s, int color, int x, int y)
{
    if (s > 0)
    { // תנאי עצירה : אורך הצלע הוא אפס
      // צעד הרקורסיה : הצלע קטנה בשתיים, מיקומה ימינה ולמטה
      DrawSquare (s, color, x, y);
      Squares(s - 2, color, x + 1, y + 1);
    }
}

public static void ScreenSaver()
{
    Random rnd = new Random();
    int s = 0, color, x, y;
    int maxLength = Math.Min(M, N); // אורך הצלע המקסימלי לא יעלה על מספר השורות או העמודות
    s = rnd.Next(0, maxLength);
    while (s != 0)
    {
        color = rnd.Next(0, 10) + 1;
        x = rnd.Next(0, N - s) + 1;
        y = rnd.Next(0, M - s) + 1;
        Squares (s, color, x, y);
        s = rnd.Next(0, maxLength);
    }
}
```

### פתרון שאלה 4

נושא מרכזי: רשימה  
סוג השאלה: פיתוח פעולות, פיתוח תכנית

```
public static void RemoveNodeK(List<int> lst, int k)
{ // הפעולה מקבלת רשימה ומספר שלם ומוחקת מהרשימה את האיבר שמקומו k
  Node<int> pos = lst.GetFirst();
  while (pos != null && k > 1)
  {
      pos = pos.GetNext();
      k = k - 1;
  }
  if (pos != null)
  {
      lst.Remove(pos);
  }
}
```

4

```

}
public static int GetNodeK(List<int> lst, int k)
{ // k הפעולה מקבלת רשימה ומספר שלם ומחזירה את ערכו של האיבר שמקומו k
Node<int> pos = lst.GetFirst();
  while (pos != null && k > 1)
  {
    pos = pos.GetNext();
    k = k - 1;
  }
  if (pos != null)
    return pos.GetInfo();
  return 0; // יוחזר 0 במקרה שלא קיים כזה איבר ברשימה
}
public static int Length (List<int> lst)
{ // הפעולה מקבלת רשימה ומחזירה את מספר האיברים ברשימה
  int length = 0;
  Node<int> pos = lst.GetFirst();
  while (pos != null)
  {
    length++;
    pos = pos.GetNext();
  }
  return length;
}

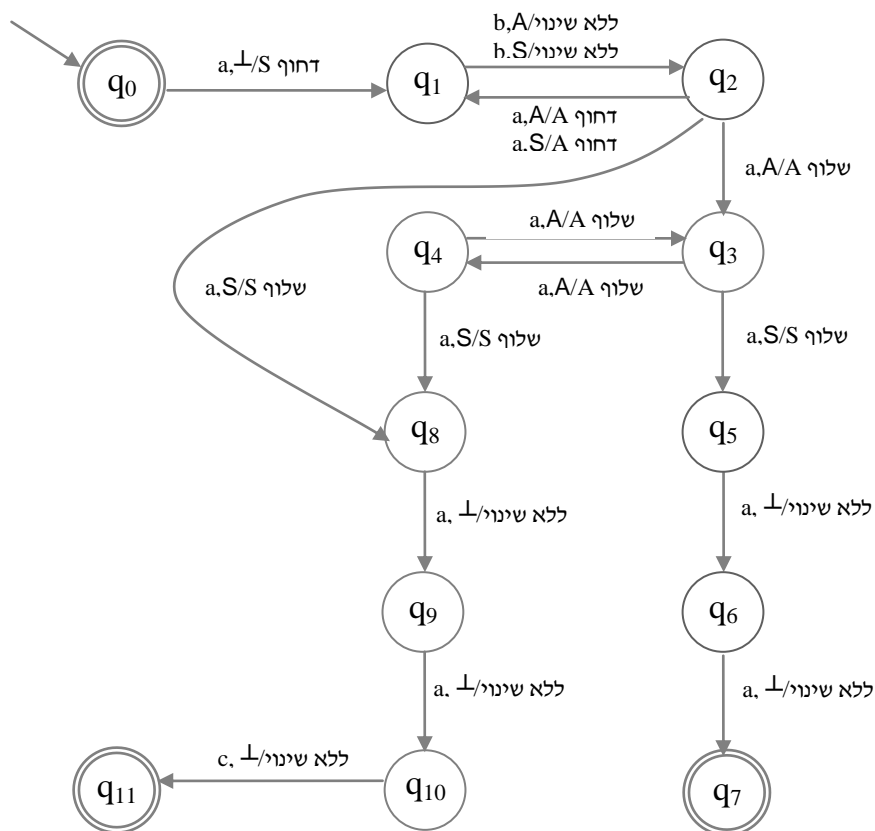
// הפעולה המבוקשת
public static List<int> Quest4 (List<int> L1, List<int> L2)
{
  List<int> L3 = new List<int>();
  Node<int> pos = L1.GetFirst();
  int len = Length(L2);
  int k;
  while (pos != null)
  {
    k = pos.GetInfo();
    if (k <= len) // יש איבר שמיקומו k ברשימה L2
    {
      if (k % 2 == 0) // זוגי k
      {
        RemoveNodeK(L2, k);
        len--;
      }
      else
      {
        L3.Insert(null, GetNodeK(L2, k));
      }
    }
    pos = pos.GetNext();
  }
  return L3;
}

```

# פתרון סדרק לני - מוראים היילובים 2006

## פתרון שאלה 13

נושא מרכזי: אוטומט מחסנית  
סוג השאלה: בניית מודל



הסבר הפתרון: תחילה נרצה לספור את מס' רצפי ה-ab שלנו, לכך מצבים  $q_0-2$ . לאחר מכן, כאשר יתחיל רצף ה-a הנוסף נרצה שגם יהיה גדול באורכו ב-2 ממספר ה-ab וגם לדעת האם זוגי או אי זוגי. ולשם כך ההפרדה בין מצבים  $q_3$  ל- $q_4$ ,  $q_3$  זוכר מספר אי זוגי של רצפים ולכן כאשר נצא ממנו עם a נוסף זה הוא יהיה מס' זוגי ואז נרצה להוסיף רק aa ואילו  $q_4$  זוכר מספר זוגי של רצפים וכאשר נצא ממנו עם a נוסף זה יהיה מס' אי זוגי ונרצה לשרשר aac. החיבור בין  $q_2$  ל- $q_8$  הוא למקרה הפרטי של  $n=1$  (הרצף abaaac).





## פתרון שאלה 14

נושא מרכזי: פעולות על מילים ושפות  
סוג השאלה: זיהוי מילים / קביעת רגולריות

### א. שייכות המילים לשפה:

המילה  $w_1=aaa$

נמצאת ב- $L_1$  כאשר  $i=3, k=0$ , מקיים את התנאי שהסכום מתחלק ב-3.  
לא נמצאת ב- $L_2$  מכיוון שמספר ה- $a$  בכל מילה של  $L_2$  הוא זוגי ואילו כאן הוא אי-זוגי.  
נמצאת ב- $L_3$  כאשר  $i=3, k=0$ , מקיים את התנאי שכל אחד מהם מתחלק ב-3.  
נמצאת ב- $L_4$  כאשר  $i=0$  ה- $a$  וה- $b$  ייעלמו וכאשר  $k=3$  יתקבל  $aaa$  ויתקיים התנאי.

המילה  $w_2=abbbaaa$

לא נמצאת ב- $L_1$ , יש  $a$  אחרי ה- $b$ , לא יכול להתקיים ב- $L_1$ .  
לא נמצאת ב- $L_2$ , ב- $L_2$  אורך רצף ה- $a$  בהתחלה חייב להיות שווה לאורך רצף ה- $a$  בסוף.  
לא נמצאת ב- $L_3$ , יש  $a$  אחרי ה- $b$ , לא יכול להתקיים ב- $L_3$ .  
נמצאת ב- $L_4$ , עבור  $k=3, i=1$ . מתחלק ב-3.

### ב. קביעה האם השפה רגולרית

השפה  $L_1$  רגולרית, מכיוון שניתן לבנות אוטומט סופי שזוכר את שארית החלוקה ב-3 של סכום האותיות. המצבים המקבלים יהיו רק אלה בהם הסכום מתחלק ב-3.

השפה  $L_2$  רגולרית. אמנם יש תלות בין מספר ה- $a$  בתחילת המילה למספר ה- $a$  בסוף המילה, אך מספר ה- $a$  הוא שארית החלוקה של  $k$  ב-3. מכיוון שיש מספר סופי של שאריות  $\{0,1,2\}$  ניתן לעשות לכל רצף  $a$  מסלול משלו, ובסה"כ 3 מסלולים עבור מספר ה- $a$ .

השפה  $L_3$  רגולרית, ניתן לזכור את שארית החלוקה, מכיוון שמספר האפשרויות סופי  $\{0,1,2\}$ , וכך אפשר לבנות אוטומט סופי עם שלושה מסלולים שונים.

השפה  $L_4$  לא רגולרית. מכיוון שישנה תלות בין ה- $a$  בהתחלה ל- $b$  נצטרך לזכור את מס' ה- $a$  בהתחלה בשביל רצף ה- $b$  שאחריו. קבוצת ההתחלות,  $W=\{\epsilon, a, a^2, a^3 \dots a^m\}$  היא אינסופית, מכיוון שישנם אינסוף התחלות ולכל התחלה נדרשת סיומת משלה. יידרשו אינסוף מצבים ולכן לא ניתן לבנות אוטומט סופי.

## פתרון שאלה 15

נושא מרכזי: פעולות על מילים ושפות  
סוג השאלה: שייכות מילים לשפה, הרכבת שפות, קביעת נכונות של טענות

א.

i.  $c$ , נמצאת ב- $L_1$  עבור  $i=1, k=0$ , ערכים כמעט מינימאליים. עבור ערכים קטנים יותר מתקבלת מילה ריקה. בנוסף  $c$  נמצאת ב- $L_3$  עבור  $i=0, k=1$ , כאשר ערכים נמוכים מהם ייתנו מילה ריקה. הבחירה היא בין מספר מופעים 0 למספר מופעים 1. 1 יבחר כחזקה המופיעה במילה פעם אחת, ו-0 יבחר כחזקה המופיעה במילה יותר מפעם אחת.

ii.  $cc$ , נמצאת ב- $L_3$  עבור  $i=0, k=2$ . כדי לקבל את המילה הקצרה ביותר 1 יבחר כחזקה המופיעה במילה פעם אחת, ו-0 יבחר כחזקה המופיעה במילה יותר מפעם אחת. בשפה  $L_2$  אורך מילה חייב להיות זוגי וגדול מ-0, כי כאשר משרשרים מילה עם ההיפוך שלה חייב להתקבל אורך זוגי.

ב. השפה  $(L_1 \cdot L_3) \cap L_2$  היא:

השלב הראשון הוא שלב השרשור:  $L_1 \cdot L_3 = \{c^k b^m a^m a^i b^j c^j | i, j, k, m \geq 0\}$   
לאחר שנעשה חיתוך עם  $L_2$ , על החלק הראשון של המילה להיות הפוך לחלק השני. הדבר יתקבל כאשר  $j=k, i=m$ , כלומר:

$$(L_1 \cdot L_3) \cap L_2 = \{c^k b^m a^m a^m b^m c^k | k, m \geq 0\} = \{c^k b^m a^{2m} b^m c^k | k, m \geq 0\}$$

ג. נכון / לא נכון

i.  $R(L_1) = L_2$

נכון,  $L_2$  פירושה שרשור מילה כלשהי עם ההיפוך שלה, המילים החדשות יהיו סימטריות (פלינדרום). היפוך כל המילים בשפה ישאיר את אותו הדבר, וזאת מעצם הגדרת הפלינדרום.

ii.  $R(L_1) = L_3$

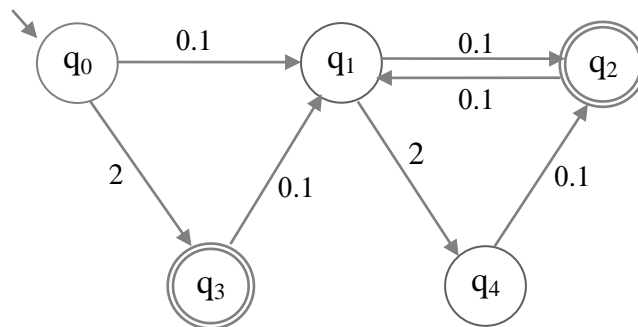
נכון, מילה בשפה  $L_3$  מכילה רצף כלשהו של  $a$ , אחריו רצף באותו האורך של  $b$  ואחריו רצף כלשהו של  $c$ . זו בדיוק ההגדרה של  $R(L_1)$ .

## פתרון שאלה 16

נושא מרכזי: אוטומט סופי דטרמיניסטי.  
סוג השאלה: בניית מודל

אוטומט סופי דטרמיניסטי שבודק אם רצף ההגרות שהתקבל על-ידי מכונת ההגרות הוא רצף חוקי. רצף הגרות חוקי מקיים את שלושת התנאים הבאים:

- הוגרלה לפחות ספרה אחת
- מספר הספרות שהוגרלו + מספר הספרות 1 שהוגרלו הוא זוגי.
- אם הוגרלה הספרה 2 היא לא הוגרלה מיד לאחר שהוגרלה ספרה 2 אחרת.



הסבר הפתרון:

על האוטומט לדאוג למצב שיזכור מספר זוגי של 0 ו-1 ומספר אי-זוגי של 0 ו-1. אסור שיהיה רצף 22, ולמעט זה אין משמעות לסדר האותיות במילה. אורך המילה לפחות אות אחת. האוטומט שהוצג הוא אוטומט לא מלא.

מספר מצב	תיאור המצב
0	מצב התחלתי
1	מספר אי-זוגי של 0,1 והאות האחרונה אינה 2
2	מספר זוגי של 0,1 והאות האחרונה אינה 2
3	מספר זוגי של 0,1 והאות האחרונה 2
4	מספר אי-זוגי של 0,1 והאות האחרונה 2