

חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"ה

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

כתיבה ועריכה:

גלית שריקי, חני טוראל

דף מספר 1



חישבו על גישה שונה לפתרון הבעיה של מיון מערך
כך שיעילותו תהיה קטנה מ- $O(n^2)$.

רמז:

היזכרו בתהליך המיזוג אותו למדתם בשנה שעברה.
כדי למיין מערך ניתן לחלקו לשני תתי מערכים שווים
בגודלם פחות או יותר, למיין כל אחד מהם, ולאחר
מכן למזג בין המערכים הממוינים.

דף מספר 2



השתמשו ב**רקורסיה** בכדי למיין מערכים,
כך שיעילות המיין שלהם תהיה פחותה מ- $O(n^2)$,
בצורה הבאה:
נחלק כל אחד מחלקי המערך לשני מערכים נוספים
עד אשר נגיע לקטע מערך באורך 1 ואז נמזג את
המערכים בצורה ממוינת
(מערך בגודל 1 הינו מערך ממוין).

דף מספר 3



לפניכם האלגוריתם למיון מיזוג, ממשו אותו.

מיון-מיזוג(arr)

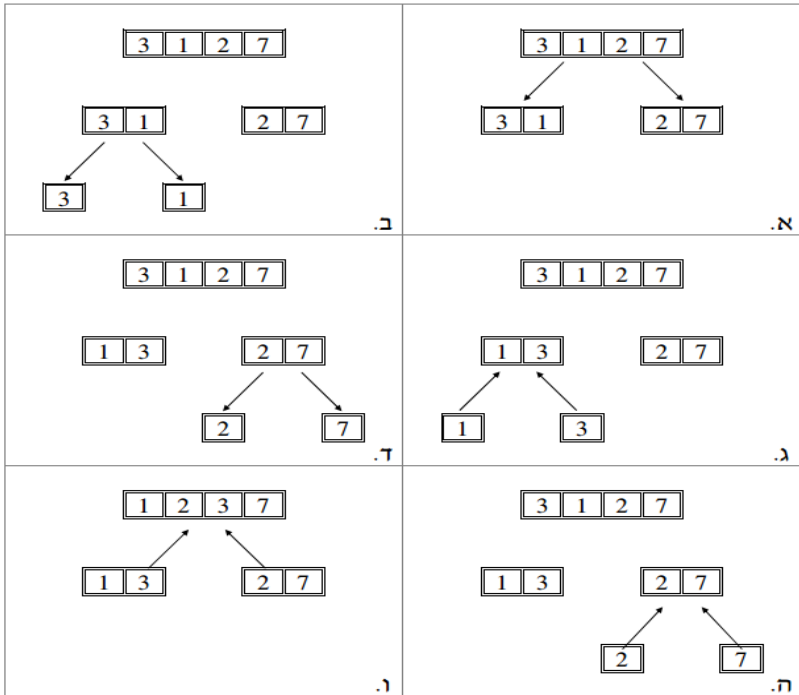
אם גודל המערך גדול מ-1 בצע:

- (1) חלק את המערך לשני תתי מערכים arr1 ו-arr2, המכילים כל אחד כמחצית מאיברי המערך
- (2) מיון-מיזוג(arr1)
- (3) מיון-מיזוג(arr2)
- (4) מזג את arr1 ואת arr2 לתוך arr

דף מספר 4



האיור הבא מתאר את ריצת האלגוריתם על מערך שבו ארבעה איברים:



ממשו את שלבים (1) ו-(4) באמצעות פעולות עזר.

דף מספר 5



לפניכם פעולה למיזוג מערכים ממוינים (שלב 4) באלגוריתם
למיון מיזוג).

ממשו את שאר הפקודות באלגוריתם.

```
public static void sort(int[] arr, int[] arr1, int[] arr2)
{
    int i=0,j=0;
    while (i < arr1.length && j<arr2.length)
    {
        if (arr1[i]<arr2[j])
        {
            arr[i+j]=arr1[i];
            i++;
        }
        else
        {
            arr[i+j]=arr2[j];
            j++;
        }
    }
    if (i < arr1.length)
        while (i < arr1.length)
        {
            arr[i+j]=arr1[i];
            i++;
        }
    else
        while (j < arr2.length)
        {
            arr[i+j]=arr2[j];
            j++;
        }
}
```

דף מספר 6



לפניכם פעולת עזר המחלקת את המערך arr לשני מערכים arr1 ו-arr2 (שלב 1) של האלגוריתם).

```
public static void divide (int[] arr, int[] arr1, int[] arr2)
{
    for (int i = 0; i < arr1.length; i++)
    {
        arr1[i]=arr[i];
    }
    for (int i = 0; i < arr2.length; i++)
    {
        arr2[i]=arr[arr.length/2+i];
    }
}
```

א. הגדירו את המערכים arr1 ו-arr2 ואת גודלם

לפני הזימון של הפעולה **divide**

ב. השלימו את שלבים (3) ו-(4) של האלגוריתם.

דף מספר 7



לפניכם האלגוריתם הכמעט מלא לפתרון הבעיה,
השלימו את הסעיפים החסרים והריצו את התכנית

```
public static void mergeSort(int []arr)
{
    if (arr.length!=1)
    {
        int[] arr1=new int[arr.length/2];
        int[] arr2=new int[arr.length-arr1.length];
        devide(arr,arr1,arr2);
        (2)
        (3)
        sort(arr,arr1,arr2);
    }
}
```


דף מספר 8



תרגיל המשך:

לפניכם האלגוריתם המוצא את המכנה המשותף הגבוה ביותר (ממ"ג) שיעילותו היא $O(n)$ שפרו את היעילות בעזרת אלגוריתם רקורסיבי.

```
public static int mamag(int a, int b)
{
    for (int i=a-1;i>1;i--)
    {
        if (b%i == 0 && a%i == 0)
            return i;
    }
    return 1;
}
```