

היום עסקנו בנושא של מיון מערך. ניסינו קודם לפתח דרך שבה נוכל להשוות את תאי המערך בזה אחר זה ולהזיז ערכים למצב שלא יהיה ערך גדול לפני קטן. הדוגמה שלנו תעסוק במערך של מספרים שלמים.

במילים אחרות, אם למשל גודל המערך - arr הוא 2, והתאים שלו מכילים את: 3,4 (כאשר: arr[0]=3 ו-arr[1]=4) המצב אינו רצוי (כי אנו מעוניינים למיין מהקטן לגדול). לכן יש להחליף בין התאים, אולם אם נכתוב פשוט:

```
arr[0]=a[1];
arr[1]=a[0];
```

מה יקרה? אחרי פקודה: a[0]=a[1]; הערך גם של: a[0] וגם של a[1] יהיה: 4 - הערך שמימין לשווה 'הורס' את הערך שמשמאל לשווה. ולכן גם בפקודה השניה זה לא ישתנה, ובסוף בשני התאים יהיה 4. לא רצוי.

ובכן, כדי להחליף ערכים בין שני משתנים (במקרה זה בין שני תאים של מערך, אבל אין בזה הבדל), אנו צריכים משתנה עזר. נניח שהגדרנו משתנה בשם: temp, כעת ניכתוב:

```
int temp;
temp=arr[0];
arr[0]=arr[1];
arr[1]=temp;
```

מה ההבדל כעת? קודם שמרנו את arr[0] ב-temp, ואחר כך 'הרסנו' את: arr[0], כאשר שמנו בו את: arr[1], כך העברנו את התוכן של arr[1] לתוך arr[0]. ולבסוף הערך שנישמר ב-temp (שהיה arr[0] המקורי) ניכנס לתוך arr[1] החדש.

במערך של 2 תאים בלבד, כמובן הכל הסתדר. מה יקרה אם נתון מערך (בגודל 5) כמו:

3,4,5,2,1

במצב כזה ניצטרך לולאה שעוברת על כולו ושואלת כל פעם על שני תאים סמוכים אם הם בסדר הנכון (ואם הם לא, נחליף). בצורה מעין זו:

```
for(int i=0; i<(5-1);++i)
{
    if (arr[i] > arr[i+1] )
    {
        temp=arr[i];
        arr[i]=arr[i+1];
        arr[i+1]=temp;
    }
}
```

שימו לב שכתבתי 5-1, כהגבלה על הלולאה כלומר, גודל המערך פחות 1. מדוע עשיתי זאת? כי בתוך הלולאה אני משווה שני תאים סמוכים ומתייחס אליהם בעזרת האינדקסים i ו- i+1. אם הייתי נותן ללולאה לרוץ עד i=4, הרי ש: i+1 היה שווה ל-5 ואז היינו מקבלים שגיאה, כי המערך מכיל תאים מ- a[0] ועד a[4] (אז אין a[5]).

בכיתה הראיתי איך לעשות את הלולאה עם התחלה מ i=1 וההשוואה הייתה: (arr[i-1] > arr[i]) . זהו בדיוק אותו רעיון, כי הלולאה צריכה לרוץ אחד פחות מגודל המערך.

זה גם הגיוני שאנחנו משוים עד שהגענו לאחד לפני אחרון עם אחרון. אם היו 5 תאים, ביצענו 4 השוואות. מה קרה כתוצאה מהלולאה הפעם? קיבלנו את המערך החדש:

3,4,2,1,5

כתוצאה קיבלנו מערך, כאשר התא האחרון (בדוגמה עם מציין 4) יש בו את הערך הגדול ביותר. הגדול ביותר בכל פעם מתחלף עד שמגיע לסוף.

אמרנו גם שאם נחזור על התהליך פעם שניה (מעבר על כל המערך שוב עם השוואות והחלפות), הפעם יסתדר הערך הבא בגודלו בדיוק במקום הרצוי (אחד לפני אחרון), כי הוא יתחלף עם כולם חוץ מהאחרון. אם נמשיך שוב ושוב כך, לבסוף המערך כולו יהיה ממויין.

זהו בעצם מקרה טיפוס שדורש לולאה בתוך לולאה. לכן 'עטפנו' את הלולאה הנ"ל בעוד לולאה שרצה מספר פעמים כגודל המערך, בצורה הבאה:

```
for ( int j=0; j< arr.Length; ++j )
{
    for( int i=0; i<arr.Length-1; ++i )
    {
        if ( arr[i] > arr[i+1] )
        {
            temp = arr[i];
            arr[i] = arr[i+1];
            arr[i+1]= temp;
        }
    }
}
```

שימו לב שהשתמשתי באינדקס בשם j עבור הלולאה החיצונית. השתמשנו גם במאפיין של מערך בשם Length שתמיד נותן את אורכו ואמרנו שזה יכול לחסוך אם נרצה לשנות את גודל המערך, אפשר רק לשנות את ההגדרה, ובכל מקום שיופיע שיפיע arr.Length הוא יקח בצורה אוטומטית את אורכו המדוייק של arr.

תרגילי מחשבה

- 1) חישבו כיצד ניתן לשפר את השיטה, כך שהלולאה הפנימית לא תמיד צריכה לרוץ על כל גודל המערך פחות אחד.
- 2) חישבו על דרך שבה אפשר עוד לקצר את התהליך, למשל אם המערך כבר ממויין מראש (או שהתמיין מהר), האם אפשר למצוא שיטה שבה התוכנית תבין מהר שהוא ממויין ותפסיק לעבוד כל כך קשה? כמו שהיא עכשיו על מערך של 10 תאים היא תצטרך לעשות בערך 100 השוואות (10x10)

לדוגמה אם המערך המקורי היה:

1
2
4
6
8
7

אחרי מעבר אחד בלולאה החיצונית הוא כבר יהיה ממויין (לכן לא צריך 36 השוואות). איך אפשר 'להסביר' זאת לתוכנית?