

סיכום שיעור

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. הדוגמא 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 ביניהם, אולם אם נכתוב פשוט:

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

מה יקרה? אחרי פקודה: arr[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 בן 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]

במערך של 2 תאים 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])

זה גם הגיוני שאנחנו משוים עד שהגענו לאחד לפני אחרון עם אחרון. מה קרה כתוצאה מהלולאה הפעם? קיבלנו את המערך החדש:

3,4,2,1,5

המערך הסתדר חלקית. מה שבטוח זה שהערך הגדול (הכבד) ביותר יהיה בטוח בסוף. כלומר שיטה זו גורמת לפחות לערך אחד להגיע למקומו. אם נריץ שוב את הלולאה הנ"ל על המערך החדש, נקבל:

3,2,1,4,5

כעת ה-4 הגיע למקומו הטבעי (אחד מהסוף), בקיצור ניצטרך מספר הרצות של אותה לולאה. כמה כאלה? כמספר התאים פחות אחד, כי אם סידרנו 4 תאים, החמישי ממילא יהי כבר מסודר.

איך מריצים לולאה מספר פעמים? ניחשתם נכון: לולאה בתוך לולאה, התוכנית הסופית תיראה כך:

```
for (int j=0; j<4; ++j)
{
  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 שהמשתנה של הלולאה החיצונית: j אינו משמש כאינדקס, אלא רק סופר את מספר הפעמים שמבצעים את הלולאה הפנימית. מיון זה ניקרא 'מיון בועות', כי בכל פעם שרצה הלולאה החיצונית, הערך 'כבד' ביותר, שוקע ל'תחתית' המערך, עד שמתחתיו יש רק ערכים כבדים ממנו.

הצענו גם שיפור שיחוסך לנו זמן. אם נריץ במקרה הנ"ל הלולאה החיצונית תתבצע 4 פעמים והפנימית 4 על כל פעם של החיצונית, כלומר בסך הכל $4 \times 4 = 16$ פעמים הפעלת ה-`if`. אמרנו שאם עוברים על המערך ואין שום החלפה, הרי הוא כבר ממוין. למשל, בניח שיש מערך שנראה כך:

3,7,9,12

נסו לעבור עליו אם פקודת ה-`if` ותראו שלא התבצעה שום החלפה של תאים.

לכן, נגדיר איזה דגל (משתנה בולאני ניקרא לו `change`) שיתחיל כ-`true` נהפוך אותו ל-`false` לפני תחילת לולאה הפנימית ואם תהיה החלפה הוא יהפוך ל-`true` שוב. בלולאה החיצונית נוסיף את התנאי שהיא תמשיך או כאשר: `change` הוא `true` או שעד שנגיע לגודל המערך פחות 1. זה אומר שהלולאה החיצונית תהיה לולאת `while`.

נסו בבקשה להשלים את התוכנית היעילה יותר בבית.