<u>הרצאה 3: אלגוריתמי מיון פשוטים</u>

בעיית המיון היא אחת הבעיות החשובות והנחקרות ביותר בכל מדעי המחשב. בהרצאה זו נעסוק באלגוריתמי מיון פשוטים שסיבוכיותם $\Theta\left(n^2\right)$. במהלך ההרצאה נציג שלושה אלגוריתמי מיון קלאסיים:

1. מיון הכנסה (Insertion Sort).
2. מיון בחירה (Selection Sort).
3. מיון בועות (Bubble Sort) .

לאחר דיון קצר בהוכחת נכונות של אלגוריתמים למיון, נוכיח את נכונות מיון הכנסה ונתח את הסיבוכיות שלו.

בחלק השני של ההרצאה נציג שתי גרסאות של מיון בועות (Bubble Sort) , נוכיח את נכונותן וננתח את סיבוכיות הזמן שלהם במקרה הגרוע ביותר.

<u>הגדרת בעיית המיון</u>

קלט: מערך A בן n איברים (כגון מספרים) המסודרים בסדר **שרירותי**

פלט: מערך B ובו איברי A מסודרים **לפי גודלם.**

אנו מניחים אם כן שאברי המערך A משתייכים לקבוצה כלשהי שכל איבריה **סדורים**. במלים אחרות: אנו מניחים כי בין כל שני אברים a ו-b מן הקבוצה מתקיימת בדיוק אחת מן האפשרויות הבאות:

1. $a < b$
2. $a = b$.
3. $a > b$.

<u>מיון הכנסה (Insertion Sort)</u>

באלגוריתם זה, מעבירים איבר אחר איבר, ממערך הקלט אל מערך הפלט, כאשר מערך הפלט תמיד **ממוין**. בכל פעם שמטפלים באיבר תורן, **מכניסים (insert)** אותו למקומו הנכון במערך הפלט ומסיתים במקום אחד את כל אברי מערך הפלט הגדולים מן האיבר המוכנס.

מיון הכנסה - הקוד

for $i = 1$ **to** n **do**

/* מקום $A[i]$ ב $B[1],\ldots,B[i-1]$ */

$j \leftarrow i$

while $j > 1$ **and** $B[j-1] > A[i]$

do

$B[j] \leftarrow B[j-1]$

$j \leftarrow j-1$

end while

/* הכנס את $A[i]$ למקומו */

$B[j] \leftarrow A[i]$

end for

דוגמא

$i=$		1		2		3		4
4		4		4		4		
6		6		6		6		
3		3		3		3		
5		5		5		5		
1		1		1		1	8	
7		7		7	8	7	7	
8		8	8	8	7	8	2	
2	2	2	2	2	2	2	1	

$i=$		5		6		7		8
4		4		4		4	8	
6		6		6	8	6	7	
3		3	8	3	7	3	6	
5	8	5	7	5	6	5	5	
2	7	2	5	2	5	2	4	
7	5	7	3	7	3	7	3	
8	2	8	2	8	2	8	2	
1	1	1	1	1	1	1	1	

הוכחת נכונות של אלגוריתם

כדי להוכיח נכונות של אלגוריתם עלינו להוכיח:

1. האלגוריתם מסתיים בכל ריצה עם קלט חוקי.
2. בכל ריצה עם קלט חוקי, הפלט נכון.

כדי לענות על שתי הדרישות הללו, עלינו להתיחס במדויק אל פעולת האלגוריתם כולל:

1. מספר המעברים בכל לולאה.
2. ערכי המשתנים בכל שלב של החישוב.
3. הצלחה או כישלון בביצוע טסטים כגון **if** ו **while**.

מיון הכנסה - הוכחת סיום

כמו במקרים רבים, הוכחת הסיום של האלגוריתם היא מיידית ונובעת מכך שהאלגוריתם מכיל שתי לולאות מקוננות ושמספר המעברים בכל לולאה חסום על ידי גודל הקלט n.

הערה (לימוד עצמי)

אפשר להוכיח כי סיבוכיות האלגוריתם היא $\Theta(n^2)$, כלומר, בכל ריצה עם קלט שארכו n מספר הצעדים חסום על ידי cn^2, עבור איזשהו קבוע c. מכאן נובע מיידית כי האלגוריתם מסתיים.

הוכחת נכונות פלט (תזכורת)

בכל הוכחת נכונות, עלינו להתיחס לפעולת האלגוריתם על הנתונים.
במירב המקרים, האלגוריתם מכיל לולאה, ואז אפשר להוכיח טענות על ערכי המשתנים בתחילת או בסוף ביצוע לולאה.
בכל הוכחה כזאת, יש לבחון את פעולת האלגוריתם על ערכי המשתנים **על ידי התייחסות מדויקת אל תיאור האלגוריתם.**

איך מוכיחים נכונות אלגוריתמי מיון?

הגדרה: יהי A מערך בן n איברים. המערך B הוא **תמורה של** A אם קבוצת איברי B שווה לקבוצת איברי A. כלומר, אם B מתקבל מA על ידי שינוי בסדר האיברים.
כדי להוכיח נכונות פלט של אלגוריתם מיון עלינו:

1. להוכיח כי מערך הפלט הוא **תמורה** של מערך הקלט.
2. להוכיח כי מערך הפלט **ממוין**.

שימו לב: הרבה פעמים הוכחת טענות אלה היא טריויאלית ואינה דורשת התעמקות רבה.

נכונות מיון הכנסה

עלינו להוכיח כי הפלט B הוא תמורה של הקלט A וכי הפלט ממוין.
נפרק את ההוכחה לשתי טענות:

טענה 1

הפלט הוא תמורה של הקלט.

טענה 2

לכל $1 \le i \le n$ מתקיים: לאחר ביצוע המעבר ה-i בלולאה, האיברים $B[1],...,B[i]$, ממוינים בסדר עולה.

שימו לב:

נכונות האלגוריתם נובעת מיידית מטענה 1 ומטענה 2 עבור $i = n$.

הוכחת טענה 1

בכל שלב של האלגוריתם **מעבירים** איבר יחיד מA אל B. לאחר n שלבים, כל איברי A עברו אל B.

מש"ל

הוכחת טענה 2

טענה זו היא מסוג **האינוריאנטה (invariant)**. בטענה מסוג זה, טוענים כי תכונה מסוימת מתקיימת לאחר כל מעבר בלולאה.

הוכחת טענה 2 (המשך)

הוכחת נכונות של טענה כזאת, צריכה להתבסס על ניתוח של פעולת הקוד על הקלט. אם נבחן את פעולת האלגוריתם, נגלה כי הטענה בודאי נכונה לאחר מעבר אחד.

הוכחת את נכונות הטענה לאחר המעבר השני, יכולה להתבצע בהסתמך על נכונות הטענה לאחר המעבר הראשון.

באופן כללי, לכל $1 \le i \le n$, הוכחת נכונות הטענה לאחר i מעברים, מסתמכת על נכונות הטענה לאחר $i-1$ מעברים.

הוכחת הטענה (המשך)

מכאן נסיק כי כדאי להוכיח את הטענה באינדוקציה על מספר המעברים בלולאה החיצונית של האלגוריתם, כלומר באינדוקציה על i :

בסיס: $(i=1)$

במעבר הראשון בלולאת ה **for**, ערך המשתנה i , כלומר 1, מושם למשתנה j. לאחר מכן, תנאי הכניסה ללולאת ה **while** נבדק ומאחר שערך j הוא 1, הלולאה אינה מתבצעת אפילו פעם אחת וערך $A[1]$ מושם ל $B[1]$.

מש"ל

צעד האינדוקציה

נניח נכונות עבור $i-1$, כלומר נניח כי לאחר $i-1$ המעברים הראשונים בלולאת ה **for**, איברי המערך $B[1],...,B[i-1]$ ממוינים בסדר עולה.

בתחילת המעבר ה i בלולאה, ערך המשתנה j נקבע לi. לאחר מכן במהלך המעבר בלולאה הפנימית איברי המערך B, מועתקים כלפי מעלה, אחד, אחד, עד שמתמלא תנאי היציאה מן הלולאה. נתבונן במקרים הבאים, המתאימים לשני תנאי היציאה:

מקרה 1: האיבר המוכנס $A[i]$ קטן מ$B[1]$, (ולכן גם מ $B[2],...,B[i-1]$). במקרה זה, הלולאה הפנימית מתבצעת בדיוק $i-1$ פעמים. עם תום המעבר ה $i-1$, האיברים $B[1],...,B[i-1]$ מוסטים אל $B[2],...,B[i]$. איברים אלה נשארים ממוינים, וכולם גדולים מ$A[i]$.

מייד לאחר מכן, $A[i]$, מושם ל$B[1]$. מהנחת האינדוקציה ומתנאי המקרה, נובע כי בסיום ביצוע המעבר הi , איברי המערך $B[1],...,B[i]$ ממוינים.

מש"ל

צעד האינדוקציה (המשך)

מקרה 2: האיבר המוכנס $A[i]$ גדול או שווה מ$B[1]$

יהי j_0 האינדקס המכסימלי המקיים $B[j_0] \le A[1]$ לפני תחילת ביצוע הלולאה ה i. האינדקס j_0 מוגדר היטב, כי $B[1] \le A[i]$. ביציאה מן הלולאה הפנימית, ערך המשתנה j הוא $j_0 + 1$. במהלך ביצוע הלולאה, $B[j_0 + 1],...,B[i-1]$ מוסטים ב- 1 כלפי מעלה.

ההוראה האחרונה בביצוע הלולאה החיצונית מעבירה את $A[i]$ מקום אחד מעל $B[j_0]$ והמערך B נשאר ממוין.

מש״ל

סיבוכיות מיון הכנסה

בכל ביצוע מבצעים n איטרציות.

באיטרציה ה-i מכניסים את האיבר $A[i]$ למקומו בחלק הממוין של B. כדי לעשות זאת, מסיטים את איברי B, הגדולים מ$A[i]$, ב1 כלפי מעלה.

במקרה הגרוע ביותר, כאשר A ממוין בסדר **הפוך**, באיטרציה ה-i יש להסיט בדיוק $i-1$ איברים והיא דורשת ביצוע של $\Theta(i)$ צעדים. מכאן, הסיבוכיות הכללית מתקבלת על ידי הטור החשבוני

$$T(n) = \sum_{i=1}^{n} i = \Theta(n^2)$$

מיון בחירה (Selection Sort)

מיון בחירה הוא עוד אלגוריתם מיידי למיון:

האלגוריתם פועל בn איטרציות. בכל איטרציה, סורקים את הקלט, בוחרים את האיבר המינימלי מבין האיברים שאינם מסומנים (בתחילת הביצוע אף איבר אינו מסומן), מעבירים אותו למקום הפנוי הראשון במערך הפלט ומסמנים אותו במערך הקלט.

תרגיל

1. כתוב קוד דמה לאלגוריתם.
2. הוכח את נכונות האלגוריתם.
3. מהי סיבוכיות האלגוריתם?

מיון בועות (Bubble Sort)

מיון בועות הוא אלגוריתם קלאסי ששנים ארוכות נחשב כאלגוריתם המיון היעיל ביותר הקיים.

יתרונו הגדול של האלגוריתם הוא **קלות הקידוד** וחסרונו הגדול, כפי שנוכיח, הוא **סיבוכיות זמן גבוהה מדי.**

תחילה נציג אלגוריתם בסיסי, נוכיח את נכונותו וננתח את הסיבוכיות שלו. נמשיך בשיפור האלגוריתם וננתח את הסיבוכיות הממוצעת של האלגוריתם המשופר.

היפוכים

היפוך (Inversion) במערך A, הוא זוג ערכים $A[i], A[j]$ i, j, המקיים: $i < j$ אך $A[i] > A[j]$.

שימו לב: מערך המכיל היפוך **אינו ממוין**.

במהלך האלגוריתם, המתואר בשקף הבא, האלגוריתם בוחן זוגות איברים צמודים ובכל פעם שמתגלה היפוך, האלגוריתם **מחליף** (Swap) בין שני האיברים שנבדקו.

מהלך האלגוריתם

חזור עד שבA אין יותר היפוכים:

השווה את אברי המערך זוג אחר זוג, החל מן האיבר הראשון, לפי הסדר הבא: $(A[1], A[2])$, $(A[2], A[3])$, ... , $(A[n-1], A[n])$.

בכל פעם שתתקל בהיפוך, למשל באיברים $A[i], A[i+1]$, החלף בין $A[i]$ לבין $A[i+1]$.

לאחר מכן חזור על אותה פעולה עצמה אך עצור לאחר השוואת האיברים $A[n-2]$ עם $A[n-1]$, וכן הלאה.

באיטרציה האחרונה השווה אך ורק את $A[1]$ עם $A[2]$.

האלגוריתם הבסיסי

```
BaseBubble(A)
for LIM = n downto 2 do
    for j = 1 to LIM - 1 do
        if A[j] > A[j+1]
            Swap(A[[j], A[j+1])
        end if
    end for
end for
```

השגרה $Swap$ מקבלת שני אברים בזכרון ומחליפה ביניהם.

האלגוריתם נקרא **מיון בועות** שכן בכל איטרציה האיבר המכסימלי **עולה** לראש המערך כבועה הצפה על פני המים.

הערות לאלגוריתם

האלגוריתם מתנהל ב $n-1$ איטרציות.

בתחילת כל איטרציה, המערך A מחולק לחלק **עליון ממוין** ולחלק **תחתון שמצבו (מבחינת המיון) אינו ידוע,** ומתקיים: כל איבר בחלק העליון **גדול ממש** מכל איבר בחלק התחתון.

המשתנה LIM מצביע על האיבר העליון בחלק התחתון.

בכל איטרציה, עובר איבר אחד מן החלק התחתון אל החלק העליון וערך המשתנה LIM קטן ב 1.

הוכחת נכונות

האלגוריתם מכיל שתי לולאות מקוננות, מספר המעברים בכל לולאה חסום על ידי n ולכן ברור כי האלגוריתם מסתיים. בכל מהלך ביצוע האלגוריתם, איברי המערך מוזזים אך ורק על ידי החלפה בין שניים מהם ועל כן ברור כי מערך הפלט הוא תמורה של מערך הקלט. נותר לנו להוכיח אך ורק כי מערך הפלט ממוין.

לכל $1 \le i \le n$ נסמן ב M_i את האיבר ה i בגודלו בA, כלומר M_1 *הוא* האיבר המכסימלי ב A, M_2 האיבר השני בגודלו בA וכן הלאה.

הוכחת נכונות (המשך)

נכונות האלגוריתם מוכחת על ידי הטענה הבאה:

טענה

לכל $1 \le i \le n$, לאחר האיטרציה ה i מתקיים:

$A[n]=M_1,\ldots,A[n-i+1]=M_i$.

שיויונים אלה נשמרים עד סוף ביצוע האלגוריתם.

להלן יקראו אברים אלה (כולל M_i) בשם: **החלק הממוין** של המערך A.

החלק השני של המערך A:

$A[1],A[2],\ldots,A[n-i]$

יקרא: **אזור אי הודאות**

ההוכחה באינדוקציה על i.

הוכחת נכונות (המשך)

בסיס: ($i=1$) עלינו להוכיח כי בתום האיטרציה הראשונה, $A[n]=M_1$, והאיבר $A[n]$ לא ישתנה יותר.

יהי j האינדקס של M_1 במערך הקלט. במהלך האיטרציה הראשונה, M_1 לא זז ממקומו עד אשר האיבר $A[j]$ מושווה עם $A[j+1]$ ואז הם מוחלפים. מרגע זה ועד תום האיטרציה, האיבר M_1 מושווה עם כל איבר הנמצא מעליו ומאחר שהוא האיבר המכסימלי בA M_1, מחליף את מקומו בכל השוואה, ובסוף האיטרציה מתקיים $A[n]=M_1$.

הוכחת נכונות (המשך)

מן הקוד נובע כי באיטרציות הבאות, האיבר $A[n]$ לא יושווה יותר עם אף איבר נוסף ומקומו לא ישתנה עד תום ביצוע האלגוריתם.

מש"ל

צעד האינדוקציה

נניח נכונות עבור $1,2,\ldots,i$ ונוכיח עבור $i+1$. לפי הנחת האינדוקציה, בתחילת האיטרציה ה -$i+1$, המשתנה LIM מצביע על האיבר $A[n-i+1]$, האיברים $M_1,\ldots,M_i$ מאכלסים את החלק הממויין של המערך A וM_{i+1} הוא האיבר המכסימלי באיזור אי הוודאות.

צעד האינדוקציה (המשך)

יהי $j \le n-(i+1)$ האינדקס של M_{i+1} בתחילת האיטרציה ה - $i+1$.
בדומה להוכחת מקרה הבסיס, M_{i+1} לא יזוז ממקומו עד שתתקיים ההשוואה בין $A[j+1]$ לבין $A[j]$ ואיברים אלה יוחלפו. לאחר מכן M_{i+1} יושווה לכל האיברים שמעליו, והוא יוחלף עם כל אחד מהם, עד שהוא יגיע, בסוף האיטרציה ה - $i+1$, אל המקום ה- $n-(i+1)$ במערך A. באותו רגע, הנחת האינדוקציה מתקיימת.

מש"ל

נכונות מיון בועות (המשך)

הוכחת הנכונות מושלמת על ידי המשפט הבא:

משפט

מיון בועות הוא אלגוריתם מיון.

הוכחה

ההוכחה נובעת מיידית מנכונות טענת האינדוקציה עבור $i = n$.

מש"ל

ניתוח סיבוכיות הזמן

בקרת התכנית, למעט הקריאות לשגרה $Swap$, אינה תלויה באברי הקלט אלא באורכו בלבד. בריצה עם מערך באורך n, מתבצעות $n-1$ איטרציות, עבור $LIM = n,,...,2$.
בכל איטרציה כזו מתבצעות $LIM-1 = n-1,...,1$ השוואות וכל השוואה גוררת לכל היותר חילוף אחד.
סך כל ההשוואות אם כן הוא

$$\sum_{1}^{n-1} i = \frac{n(n-1)}{2} = \frac{n^2-n}{2} = \Theta(n^2)$$

מכאן, סיבוכיות הזמן של האלגוריתם היא $\Theta(n^2)$.

מוטיבציה לאלגוריתם המשופר

ישנם מערכים עבורם מספר ההיפוכים קטן בהרבה מn^2. בפרט, עבור מערך **ממויין**, מספר ההיפוכים הוא 0. נכנה קלטים בהם מספר ההיפוכים **קטן** בשם **קלטים נוחים**.
האלגוריתם המשופר מנסה להוריד את זמן הריצה עבור קלטים נוחים.

דוגמא לקלט נוח:

1	5	3	2	4	7	6	8

ההיפוכים בקלט הם:

$(5,3),(5,2),(5,4)(3,2),(7,6)$

האלגוריתם המשופר

זוהי גרסה משופרת החוסכת בצעדי חישוב עבור קלטים נוחים.

$ImpBubble(A)$

$LIM \leftarrow n$

while $LIM > 1$ **do**

  $newLIM \leftarrow 1$

  for $j = 1$ to $LIM - 1$ **do**

    if $A[j] > A[j+1]$ **then**

      $Swap(A[j], A[j+1])$

      $newLIM \leftarrow j$

    end if

  end for

  $LIM \leftarrow newLIM$

end while

הערות לאלגוריתם המשופר

האלגוריתם המשופר מסתמך על העובדה שעבור קלטים **נוחים**, הגבול בין החלק ה**ממוין**, לבין **אזור אי הוודאות**, יכול לקטון בכל איטרציה ביותר מ1.

בכל איטרציה, חלק המערך העליון, עליו עוברים לאחר ביצוע החילוף האחרון, הוא **ממוין**. לעומת זאת, אין לנו כל ערובה לגבי איזור אי הוודאות.

באלגוריתם המשופר, ערך המשתנה LIM נקבע עם תום כל איטרציה, כך שיצביע על האיבר התחתון של החילוף האחרון שהתבצע באיטרציה.

דוגמא:

המשתנה LIM מצביע על התא המוצלל בהיר והמשתנה $newLIM$ מצביע על התא המוצלל כהה:

1	5	3	2	4	7	6	8

1	3	5	2	4	7	6	8

1	3	2	5	4	7	6	8

1	3	2	4	5	7	6	8

1	3	2	4	5	6	7	8

1	3	2	4	5	6	7	8

1	3	2	4	5	6	7	8

1	2	3	4	5	6	7	8

1	2	3	4	5	6	7	8

הוכחת נכונות האלגוריתם המשופר

ההוכחה הפורמלית אינה קלה.

לדוגמא: מספר האיטרציות בלולאה החיצונית, אינו קבוע מראש והוא תלוי בקלט. לכן עלינו להוכיח:

טענה

האלגוריתם מסתיים.

הוכחה

בתחילת האלגוריתם, ערך המשתנה LIM נקבע לn. בכל מעבר בלולאה, ערך המשתנה LIM יורד לפחות ב-1. מכאן, מספר האיטרציות בלולאה החיצונית הוא לכל היותר $n-1$.

הוכחת נכונות (המשך)

ברור כי הפלט הוא תמורה של הקלט ולכן כל מה שצריך להוכיח הוא כי

טענת האינדוקציה: יהי LIM_i ערכו של המשתנה LIM בתום איטרציה i ויהי ה **החלק העליון** של A, תת המערך הנמצא מעל LIM_i. תחת הגדרות אלה, בתום איטרציה i מתקיים:

איברי A הם תמורה של איברי הקלט, **החלק העליון** ממוין, כל אבריו גדולים מכל איברי אזור אי הוודאות.

תרגיל: הוכח את הטענה פורמלית באינדוקציה על מספר המעברים (איטרציות) בלולאת האלגוריתם.

סיבוכיות הזמן באלגוריתם המשופר

באלגוריתם המשופר, זמן הריצה תלוי במספר ההיפוכים בקלט ולא רק באורכו:

1. עבור קלט ממוין בסדר עולה, סיבוכיות הזמן היא **לינארית.** זהו המקרה הטוב ביותר.
2. סיבוכיות הWorst Case מתקבלת על ידי קלט בממוין בסדר יורד. במקרה זה, באיטרציה ה i מתבצעים בדיוק $n-i$ חילופים ובסך הכל, מספר החילופים הכולל הוא , כמו באלגוריתם הקודם, $O(n^2)$. אפשר להראות כי זהו גם מספר הצעדים המכסימלי שהאלגוריתם מבצע עבור קלט כלשהו.

סיכום

בהרצאה זו הצגנו שלושה אלגוריתמי מיון קלאסיים:

1. מיון הכנסה (Insertion Sort).
2. מיון בחירה (Selection Sort).
3. מיון בועות (Bubble Sort) .

לכל אלגוריתם, הוכחנו את נכונותו ונתחנו את הסיבוכיות שלו במקרה הגרוע ביותר. לאכזבתנו, סיבוכיות שלושת האלגוריתמים הללו היא

$$\Theta(n^2)$$