<u>סדרי גודל (אסימפטוטיים)</u>

דרך לתאר זמן ריצה / כמות זיכרון של תוכנית מחשב.
לא מתעניינים במדדים מדויקים אלא <u>בסדרי גודל</u>.

שלוש מחלקות:

1. $O(\cdot)$ במשמעות של "≤"
2. $\Omega(\cdot)$ במשמעות של "≥"
3. $\Theta(\cdot)$ במשמעות של "="

באופן פורמלי, אם $g(n)$ היא פונקציה המוגדרת מעל הטבעיים אז

1. מחלקת הפונקציות $O(g(n))$ כוללת כל פונקציה $f(n)$ עבורה קיימים קבועים c,n_0 כך ש
$f(n) \le c \cdot g(n) \quad \forall n \ge n_0$.
2. מחלקת הפונקציות $\Omega(g(n))$ כוללת כל פונקציה $f(n)$ עבורה קיימים קבועים c',n_0 כך ש
$f(n) \ge c' \cdot g(n) \quad \forall n \ge n_0$.
3. מחלקת הפונקציות $\Theta(g(n))$ כוללת כל פונקציה $f(n)$ עבורה קיימים קבועים c,c',n_0 כך ש
$c' \cdot g(n) \le f(n) \le c \cdot g(n) \quad \forall n \ge n_0$.

שיטת סימון: בדרך כלל, כשנרצה לומר שהפונקציה $f(n)$ נמצאת במחלקה $O(g(n))$, נכתוב $f(n) = O(g(n))$. כנ"ל לגבי המחלקות $\Omega(g(n))$ ו $\Theta(g(n))$.

<u>טענה</u>
$f(n) = \Theta(g(n))$ אם ורק אם $f(n) = O(g(n))$ וגם $f(n) = \Omega(g(n))$.
<u>הוכחה</u>
כיוון 1: נניח ש $f(n) = \Theta(g(n))$. לפי ההגדרה, קיימים קבועים c,c',n_0 כך שלכל $n \ge n_0$, מתקיים $f(n) \le c \cdot g(n)$ וגם $f(n) \ge c' \cdot g(n)$. מכאן, נוכל לבחור בקבועים c,n_0 ולקבל $f(n) = O(g(n))$ ובנפרד, לבחור בקבועים c',n_0 ולקבל $f(n) = \Omega(g(n))$.■
כיוון 2: נניח ש $f(n) = O(g(n))$ וגם $f(n) = \Omega(g(n))$. לפי ההגדרות קיימים קבועים c,n_0 כך ש $f(n) \le c \cdot g(n) \quad \forall n \ge n_0$ וגם קיימים קבועים c',n'_0 כך ש $f(n) \ge c' \cdot g(n) \quad \forall n \ge n'_0$.
נגדיר $m_0 = \max\{n_0, n'_0\}$ ונקבל באמצעות בחירה בקבועים c,c',m_0 ש $f(n) = \Theta(g(n))$.■

ההגדרה הנ"ל היא *ההגדרה הסטנדרטית* למחלקות $O(\cdot),\Omega(\cdot),\Theta(\cdot)$. קיימת *הגדרה אלטרנטיווית*:
עבור פונקציות $f(n), g(n)$ המוגדרות מעל הטבעיים:

1. $f(n) = O(g(n))$ אם קיימים קבועים a,b כך ש $f(n) \le a + b \cdot g(n)$ לכל n.
2. $f(n) = \Omega(g(n))$ אם קיימים קבועים a,b כך ש $f(n) \ge a + b \cdot g(n)$ לכל n.
3. $f(n) = \Theta(g(n))$ אם $f(n) = O(g(n))$ <u>**וגם**</u> $f(n) = \Omega(g(n))$.

טענה

יהיו $f(n)$ ו $g(n)$ פונקציות מעל הטבעיים.

I. אם $f(n)=O(g(n))$, אז $f(n)+g(n)=\Theta(g(n))$.

II. אם $f(n)=\Omega(g(n))$, אז $f(n)+g(n)=\Theta(f(n))$.

במילים אחרות, בחיבור של כמה גורמים, יש להתחשב בגורם מסדר הגודל "הכי משמעותי".

הוכחה

נגדיר $h(n)=f(n)+g(n)$.

I. אם $f(n)=O(g(n))$, אז לפי ההגדרה, קיימים קבועים c,n_0 כך ש
$f(n)\le c\cdot g(n)\ \forall n\ge n_0$. מכאן, מתקיים
$\forall n\ge n_0 \quad h(n)=f(n)+g(n)\le c\cdot g(n)+g(n)$. נגדיר $d=c+1$ ונקבל
$\forall n\ge n_0 \quad h(n)\le d\cdot g(n)$, לכן באמצעות בחירה בקבועים d,n_0 מתקיים
$h(n)=O(g(n))$. בכדי להראות ש $h(n)=\Omega(g(n))$ מספיק לשים לב ש $h(n)\ge g(n)$
(ניתן לבחור בקבועים $c'=1$ ו $n_0=1$).

II. אם $f(n)=\Omega(g(n))$, אז לפי ההגדרה, קיימים קבועים c',n_0 כך ש
$f(n)\ge c'\cdot g(n)\ \forall n\ge n_0$. מכאן, מתקיים
$\forall n\ge n_0 \quad h(n)=f(n)+g(n)\ge c'\cdot g(n)+g(n)$. נגדיר $d'=c'+1$ ונקבל
$\forall n\ge n_0 \quad h(n)\ge d'\cdot g(n)$, לכן באמצעות בחירה בקבועים d',n_0 מתקיים
$h(n)=\Omega(g(n))$. בכדי להראות ש $h(n)=O(g(n))$ מספיק לשים לב ש $h(n)\le g(n)$
(ניתן לבחור בקבועים $c=1$ ו $n_0=1$).

תרגיל

קבע האם $f(n)=O(g(n))$, $f(n)=\Omega(g(n))$ או "גם וגם", כלומר $f(n)=\Theta(g(n))$.

I. $f(n)=6n+15 \quad g(n)=n^2+3n$

רמז: בחיבור של כמה גורמים מספיק להתחשב בגורם מסדר הגודל "הכי משמעותי".

תשובה: $f(n)=O(g(n))$.

הוכחה: באמצעות ההגדרה הסטנדרטית. נחפש קבועים c,n_0 מתאימים. נדרוש:

$$6n+15\le c(n^2+3n)\ \forall n\ge n_0.$$

נעביר אגפים ונכנס:

$$0\le cn^2+(3c-6)n-15\ \forall n\ge n_0.$$

מספיק לבחור $c=2$, $n_0=3$ ואז מקבלים

$$0\le 2n^2-15$$

אשר באמת מתקיים לכל $n\ge 3$.

הערה: כמובן שיש הרבה בחירות אפשריות לקבועים c,n_0.

II. $f(n)=n\sqrt{n} \quad g(n)=n^2-5$

תשובה: $f(n)=O(g(n))$.

הוכחה: באמצעות ההגדרה האלטרנטיווית. נחפש קבועים a,b מתאימים. נדרוש:

$n\sqrt{n} \le a + b(n^2 - 5)$.

נכנס:

$n\sqrt{n} \le bn^2 + a - 5$.

מספיק לבחור $a = 5$, $b = 1$ **ואז מקבלים**

$$n\sqrt{n} \le n^2 \Leftrightarrow \sqrt{n} \le n$$

אשר מתקיים לכל n.

III. $f(n) = 4\log(n^3)$ $\quad g(n) = 2\log(n)$

תזכורת: מחוקי לוגריתמים: $\log(x^y) = y\log(x)$.

תשובה: $f(n) = \Theta(g(n))$.

הוכחה: באמצעות ההגדרה הסטנדרטית.נחפש קבועים c, c', n_0 מתאימים. נדרוש:

$c' \cdot 2\log(n) \le 4\log(n^3) \le c \cdot 2\log(n)\ \forall n \ge n_0$.

נפעיל את החוק הנ"ל:

$c' \cdot 2\log(n) \le 12\log(n) \le c \cdot 2\log(n)\ \forall n \ge n_0$.

מספיק לבחור $c = 6$, $c' = 6$, $n_0 = 0$ **ואז מקבלים**

$$12\log(n) \le 12\log(n) \le 12\log(n)$$

אשר באמת מתקיים לכל $n \ge 0$.

הערה: לפעמים יותר נוח להוכיח ש $f(n) = \Theta(g(n))$ על ידי שתי הוכחות נפרדות:

$f(n) = O(g(n))$ ו $f(n) = \Omega(g(n))$.

IV. $f(n) = 20 \cdot g(n)$ $\quad g(n) = 6n + 7$

תשובה: $f(n) = \Theta(g(n))$.

הוכחה: באמצעות ההגדרה הסטנדרטית. פשוט נבחר $c = 20$, $c' = 20$, $n_0 = 0$. מקבלים

$$20(6n+7) \le 20(6n+7) \le 20(6n+7)$$

אשר באמת מתקיים לכל $n \ge 0$.

מסקנה: כפל בקבוע לא משנה את סדר הגודל.

V. $f(n) = n\log(n)$ $\quad g(n) = n^{5/4}$

כלל אצבע: $\log^p(n) << n^q$ לכל זוג קבועים p, q ו n מספיק גדול.

צידוק: נוציא log משני הצדדים ונקבל

$\log(n^q) = q\log(n)$ **לעומת** $\log(\log^p(n)) = p\log\log(n)$.

תשובה: $f(n) = O(g(n))$.

הוכחה: נחלק את שתי הפונקציות ב n ונקבל $f'(n) = \log(n)$ ו $g'(n) = n^{1/4}$.

כעת ניתן להפעיל את כלל האצבע.

<u>תרגיל</u>

מהו ערכו (המדויק) של המשתנה k כפונקציה של הפרמטר n בסיום הריצה. תן סדר גודל לזמן הריצה (כפונקציה של הפרמטר n).

הרעיון: משתמשים במשתנה k בתור counter על הפעולות האטומיות שהפונקציה מבצעת. בסוף נקבע את סדר הגודל של k בכדי לקבל את סדר הגודל של זמן הריצה. היות וכפל בקבוע לא משנה את סדר הגודל, ניתן להסתכל על סדרה (קבועה) של פעולות כעל פעולה אחת. למעשה, נסתפק בקידום המונה k פעם אחת בכל איטרציה של כל לולאה.

I.
```
void f1(int n){
        int k = 0 ;
        for (int i = 1; i ≤ n; i++)
                k++;
}
```

תשובה: הלולאה מתבצעת n פעמים לכן בסיום הריצה מתקיים $k=n$ וזמן הריצה הוא מסדר גודל $\Theta(n)$.

II.
```
void f2(int n){
        int k = 0;
        for (int i = 1; i ≤ n; i++)
                for (int j = 1; j ≤ n; j++)
                        k++;
}
```

תשובה: הלולאה החיצונית מתבצעת n פעמים. בכל איטרציה של הלולאה החיצונית, הלולאה הפנימית מתבצעת n פעמים. מכאן, בסוף הריצה מתקיים $k=n^2$ ו זמן הריצה הוא מסדר גודל $\Theta(n^2)$.

III.
```
void f3(int n){
        int k = 0;
        for (int i = 1; i ≤ n; i++)
                for (int j = 1; j ≤ i; j++)
                        k++;
}
```

תשובה: הלולאה החיצונית מתבצעת n פעמים. בכל איטרציה של הלולאה החיצונית, הלולאה הפנימית מתבצעת i פעמים. מכאן, בסוף הריצה מתקיים

$$k=\sum_{i=1}^{n} i .$$

מנוסחת טור חשבוני $k=\frac{n(n+1)}{2}$ וזמן הריצה הוא מסדר גודל $\Theta(n^2)$.

IV.
```
void f4(int n){
        int k = 0;
        int i = 1;
        while (i < n){
                k++;
```

```
        i = i * 2;
    }
}
```

תשובה: נשאל את עצמנו: כמה פעמים מתבצעת הלולאה? בסוף כל איטרציה ערכו של המשתנה i גדול פי 2 מערכו בסוף האיטרציה הקודמת. באופן כללי בסוף האיטרציה ה k, מתקיים $i = 2^k$. הריצה עוצרת כאשר מתקיים $2^k = i \ge n$. מתי זה קורה? כאשר $k \ge \log(n)$. לכן, הלולאה תתבצע $\lceil \log(n) \rceil$ פעמים ובסוף הריצה מתקיים $k = \lceil \log(n) \rceil$. זמן הריצה הוא מסדר גודל $\Theta(\log(n))$.

V.
```
void f5(int n){
    int k = 0;
    int i = n;
    while (i > 1){
        k++;
        i = i / 2;
    }
}
```

תשובה: נשאל את עצמנו: כמה פעמים מתבצעת הלולאה? בסוף כל איטרציה ערכו של המשתנה i קטן פי 2 מערכו בסוף האיטרציה הקודמת. באופן כללי בסוף האיטרציה ה k, מתקיים $i = \frac{n}{2^k}$. הריצה עוצרת כאשר מתקיים $\frac{n}{2^k} = i \le 1$. מתי זה קורה? (השווה לפתרון השאלה הקודמת.) כאשר $k \ge \log(n)$. לכן, הלולאה תתבצע $\lceil \log(n) \rceil$ פעמים ובסוף הריצה מתקיים $k = \lceil \log(n) \rceil$. זמן הריצה הוא מסדר גודל $\Theta(\log(n))$.

<u>תמצית נוסחאות</u>

- $\log(x^y) = y\log(x)$.
- $\log^p(n) << n^q$ לכל זוג קבועים p,q ו n מספיק גדול.
- $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.
- $\sum_{i=0}^{k-1} 2^i = 2^k - 1$. (בנוסחה זו אתם עשויים להעזר לפתרון תרגיל הבית).