מבני נתונים

<u>מטרת הקורס</u>

- הכרת סוגים שונים של מבני נתונים ומימושם
- בחירת מבנה נתונים לפתרון יעיל של בעיה נתונה
- ניתוח זמן ריצה של אלגוריתם
- הכרת כלים לבחינת יעילות של אלגוריתמים

ד"ר נעה לוינשטיין ©
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

מהו אלגוריתם ?

אלגוריתם:

- סדרת צעדים ההופכת קלט לפלט
- אלגוריתם חייב להיות נכון
- אלגוריתם חייב לעצור

דוגמאות: מיון רשימת מספרים, חיפוש מחרוזת, דחיסת טקסט, ...

מהו מבנה נתונים ?

אוסף של פעולות על קבוצת נתונים.

דוגמאות:

- רשימה מקושרת,
- מחסנית,
- תור,
- עץ חיפוש,
- ...

מה המטרה ?

אלגוריתם שזמן הריצה שלו מהיר

איך מתארים אלגוריתם?
מהו זמן ריצה ?

בעיה

קלט: מערך $A[1..5]$ (5 איברים)

פלט: סכום איברי המערך A

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

A[1]	A[2]	A[3]	A[4]	A[5]
6				

m
0

A[1]	A[2]	A[3]	A[4]	A[5]
6	[redacted]	[redacted]	[redacted]	[redacted]

m
6

A[1]	A[2]	A[3]	A[4]	A[5]
[redacted]	4	[redacted]	[redacted]	[redacted]

m

10

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$
		1		

m

11

A[1]	A[2]	A[3]	A[4]	A[5]
			8	

m

19

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$
[redacted]	[redacted]	[redacted]	[redacted]	2

m
21

אלגוריתם לפתרון הבעיה

Sum(A[1..5])

```
m=0
for i=1 to 5 do
    m=m+A[i]
return(m)
```

אלגוריתם לפתרון הבעיה

Sum(A[1..5])

```
1     m=0
5+1   for i=1 to 5 do
5         m=m+A[i]
1     return(m)
-----
13
```

בעיה (חדשה?)

קלט: מערך $A[1..n]$ (n איברים)

פלט: סכום איברי המערך A

אלגוריתם לפתרון הבעיה

```
Sum(A[1..n])

   m=0
   for i=1 to n do
       m=m+A[i]
   return(m)
```

אלגוריתם לפתרון הבעיה

```
Sum(A[1..n])

1     m=0
n+1   for i=1 to n do
n         m=m+A[i]
1     return(m)
```

$$2\cdot n+3$$

זמן הריצה תלוי בגודל קלט
(לא פורמאלי)

בהמשך לדוגמא: סכום איברים ברשימה.

ברשימה קצרה: החישוב מהיר.

ברשימה ארוכה: החישוב דורש יותר זמן

זמן הריצה אינו מספר אלא תלוי בגודל הקלט,
כלומר, זמן הריצה הוא

פונקציה של גודל הקלט

אלגוריתם לפתרון הבעיה

Sum(A[1..5])

1	m=0
5+1	for i=1 to 5 do
5	m=m+A[i]
1	return(m)
13	

$T(5)=13$

אלגוריתם לפתרון הבעיה

Sum(A[1..n])

1	m=0
n+1	for i=1 to n do
n	m=m+A[i]
1	return(m)

$2 \cdot n+3$

$T(n)=2 \cdot n+3$

יעילות של אלגוריתמים

<u>מהו ניתוח אלגוריתם</u>?

- הערכת יעילות האלגוריתם = הערכת כמות המשאבים שהאלגוריתם דורש
  - **זמן ריצה**: מספר צעדי החישוב (פעולות בסיסיות) ביחס לגודל הקלט
  - **מקום**: כמות הזיכרון הדרוש ביחס לגודל הקלט

המטרה: מציאת אלגוריתם בעל זמן ריצה קטן ככל האפשר

בעיה

קלט: מערך $A[1..n]$ (n איברים), איבר k

פלט: T אם k איבר במערך A אחרת F

בעיה

קלט: מערך $A[1..n]$ (n איברים), איבר k

פלט: T אם k איבר במערך A אחרת F

המחשת הפתרון

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
8

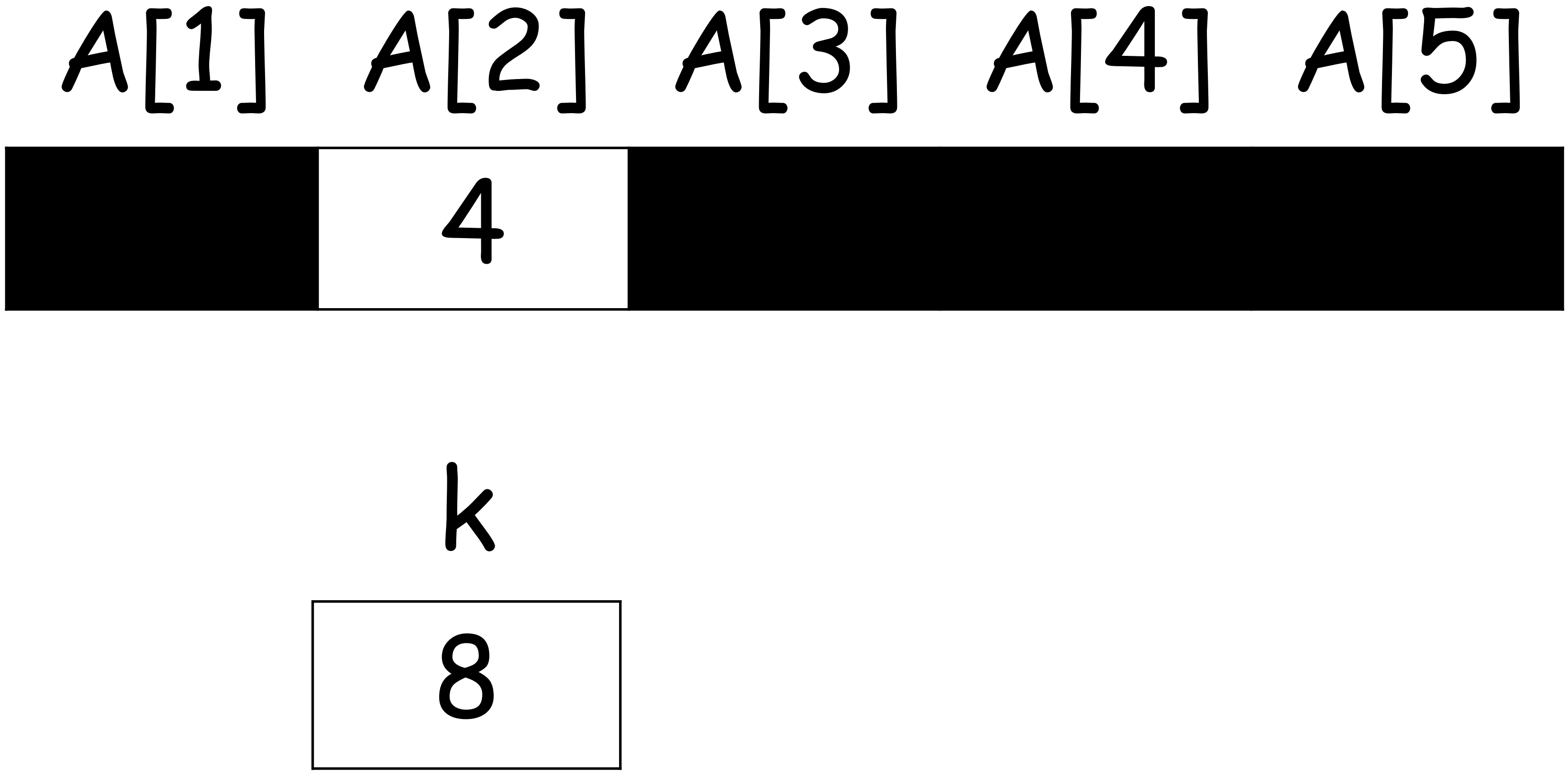
A[1]	A[2]	A[3]	A[4]	A[5]
6				

k
8

A[1]	A[2]	A[3]	A[4]	A[5]
6	[redacted]			

k

8


A[1] A[2] A[3] A[4] A[5]
4
k
8

A[1]	A[2]	A[3]	A[4]	A[5]
	4			

k

8

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$
		1		

k

8

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$
[redacted]	[redacted]	1	[redacted]	[redacted]

k

8

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$
[redacted]			8	[redacted]

k

8

A[1]	A[2]	A[3]	A[4]	A[5]
			8	

k

8

אלגוריתם לפתרון הבעיה

האם הפסאודו-קוד
חד-משמעי?

```
Find(A[1..n],k)

    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

אלגוריתם לפתרון הבעיה

```
Find(A[1..n],k)

    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

האלגוריתם לא
מסיים באותו מספר
צעדים לכל קלט

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
6

מקרה טוב

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
9

מקרה גרוע

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

מקרה ממוצע

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

מקרה ממוצע, נפריד ל-2 מקרים:

- האיבר שמחפשים נמצא במערך
- האיבר שמחפשים אינו נמצא במערך

מקרה ממוצע: האיבר שמחפשים נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k

6

מקרה ממוצע: האיבר שמחפשים נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
4

מקרה ממוצע: האיבר שמחפשים נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
1

מקרה ממוצע: האיבר שמחפשים נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
8

מקרה ממוצע: האיבר שמחפשים נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k
2

מקרה ממוצע:

האיבר שמחפשים **אינו** נמצא במערך

A[1]	A[2]	A[3]	A[4]	A[5]
6	4	1	8	2

k

9

זמן הריצה תלוי בקלט
(לא פורמאלי)

בהמשך לדוגמא: חיפוש איבר ברשימה.

האיבר בראש הרשימה.

האיבר בסוף הרשימה.

יש לאבחן בזמן הריצה בין:

המקרה הטוב

המקרה הגרוע

המקרה הממוצע

זמן ריצה

- **זמן הריצה הגרוע ביותר (worst case)**

זמן הריצה "הארוך ביותר" על איזשהו קלט בגודל n

- **זמן הריצה הטוב ביותר (best case)**

זמן הריצה "הקצר ביותר" על איזשהו קלט בגודל n

- **זמן הריצה הממוצע (average case)**

זמן הריצה הצפוי, כלומר, הממוצע בין זמני הריצה של כל הקלטים האפשריים בגודל n.

המקרה הטוב

```
Find(A[1..n],k)

1   i=1
1   While i≤n do
1       if A[i]=k return(True)
        i++
3   return(False)
```

$T(n)=3$

המקרה הגרוע

```
Find(A[1..n],k)
```

$T(n)=3\cdot n+3$

1	i=1
n+1	While i≤n do
n	if A[i]=k return(True)
n	i++
1	return(False)
$3\cdot n+3$	

המקרה הממוצע

```
Find(A[1..n],k)

    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

$$T(n)=(3+6+\ldots+3\cdot n)/n$$

האיבר נמצא במערך

```
Find(A[1..n],k)
    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

$3+6+\ldots+3\cdot n$

$$3+6+\ldots+3\cdot n$$
$$=3\cdot(1+2+\ldots+n)$$

לפי נוסחת
סכום סדרה
חשבונית:

$$(1+n)\cdot(n/2)$$

$$1+2+\ldots \qquad +n$$

$\vdots$

ד"ר נעה לוינשטיין ©
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

$$T(n)=(3+6+\ldots+3\cdot n)/n= a\cdot n+b$$

האיבר נמצא במערך

```
Find(A[1..n],k)
    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

המקרה הממוצע

```
Find(A[1..n],k)

    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

$T(n)=3\cdot n+3$

האיבר אינו נמצא במערך

```
Find(A[1..n],k)
    i=1
    While i≤n do
        if A[i]=k return(True)
        i++
    return(False)
```

לא רוצים תלות במחשב

- בכל מחשב זמן ביצוע הפקודה יכול להשתנות, אך השינוי הוא עד כדי מכפלה בקבוע (או תוספת של קבוע)
- נתעניין בזמני ריצה מבחינת "סדר גודל" ולא בזמני הריצה המדויקים.
- נתעלם מזמן הריצה על קלטים קטנים, ונתעלם ממכפלה פי קבוע.

שיעור הגידול – לא פורמאלי

- מתעלמים מהעלות במימוש הפקודות בשפת מכונה ומסמנים אותם כקבועים- $c_1,c_2,...$
- מתעלמים מהקבועים השונים ומתייחסים לסכום של קבועים שונים כקבוע אחד.
- מתעניינים רק בקצב הגידול (שיעור) לכן מתחשבים רק בחלק המשמעותי ביותר בנוסחה ומתעלמים מהשאר.

תהליך חיפוש אלגוריתם טוב ביותר

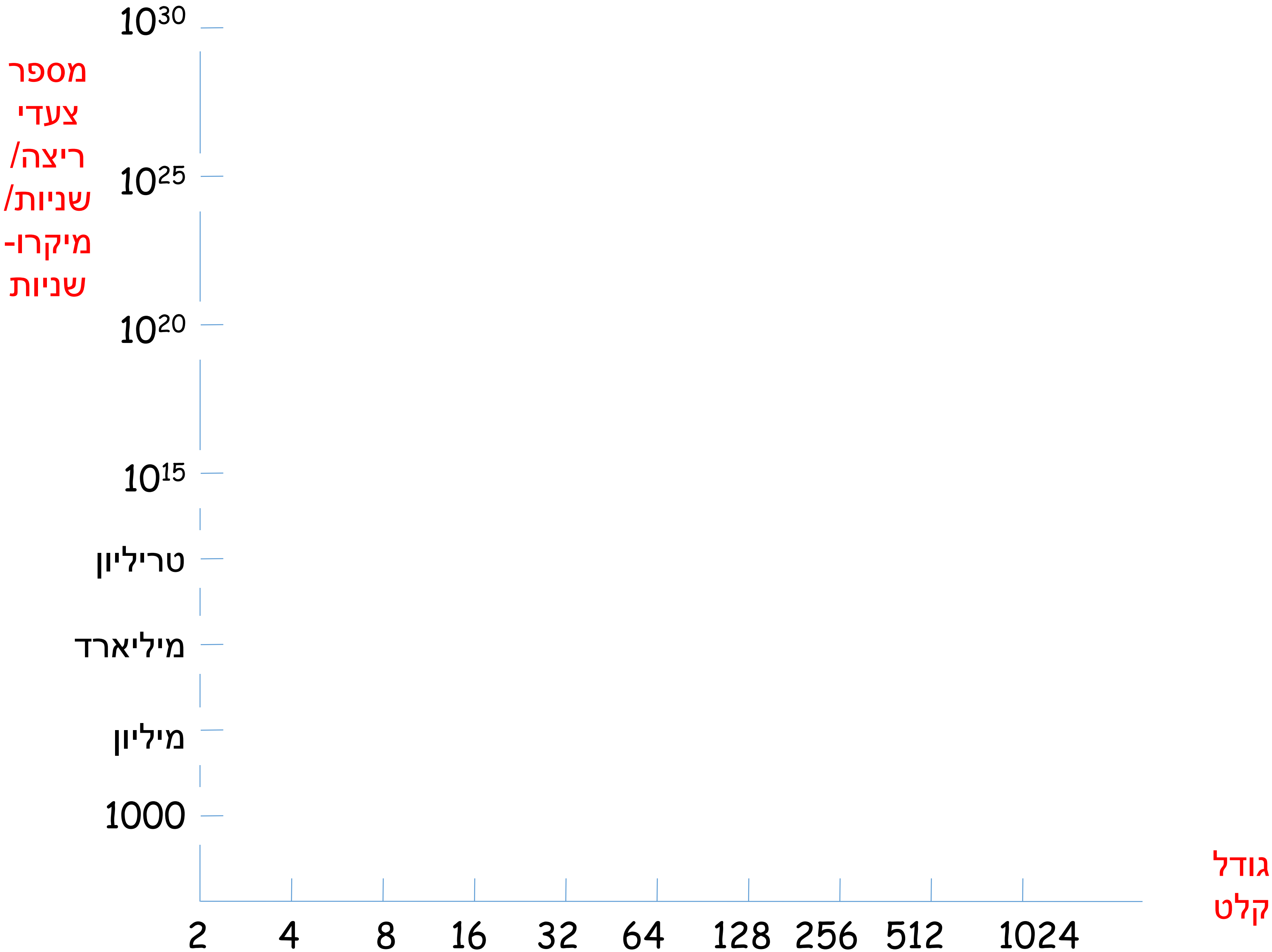
- נכתוב מספר אלגוריתמים לפתרון הבעיה.
- לכל אלגוריתם נחשב את הפונקציה המבטאת את זמן הריצה שלה.
- נשווה בין הפונקציות השונות.

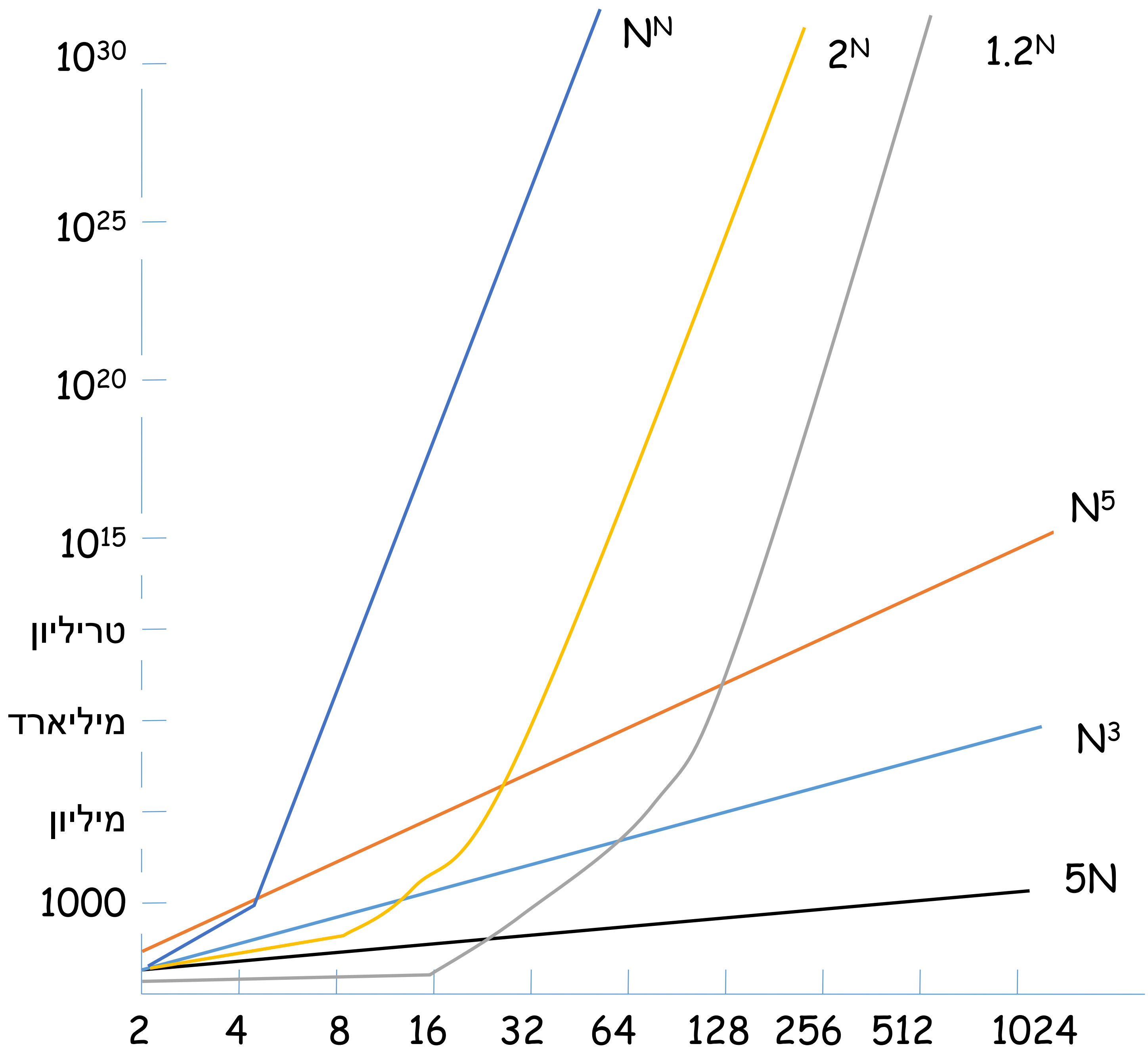
כיצד משווים בין "גודלי" פונקציות ?

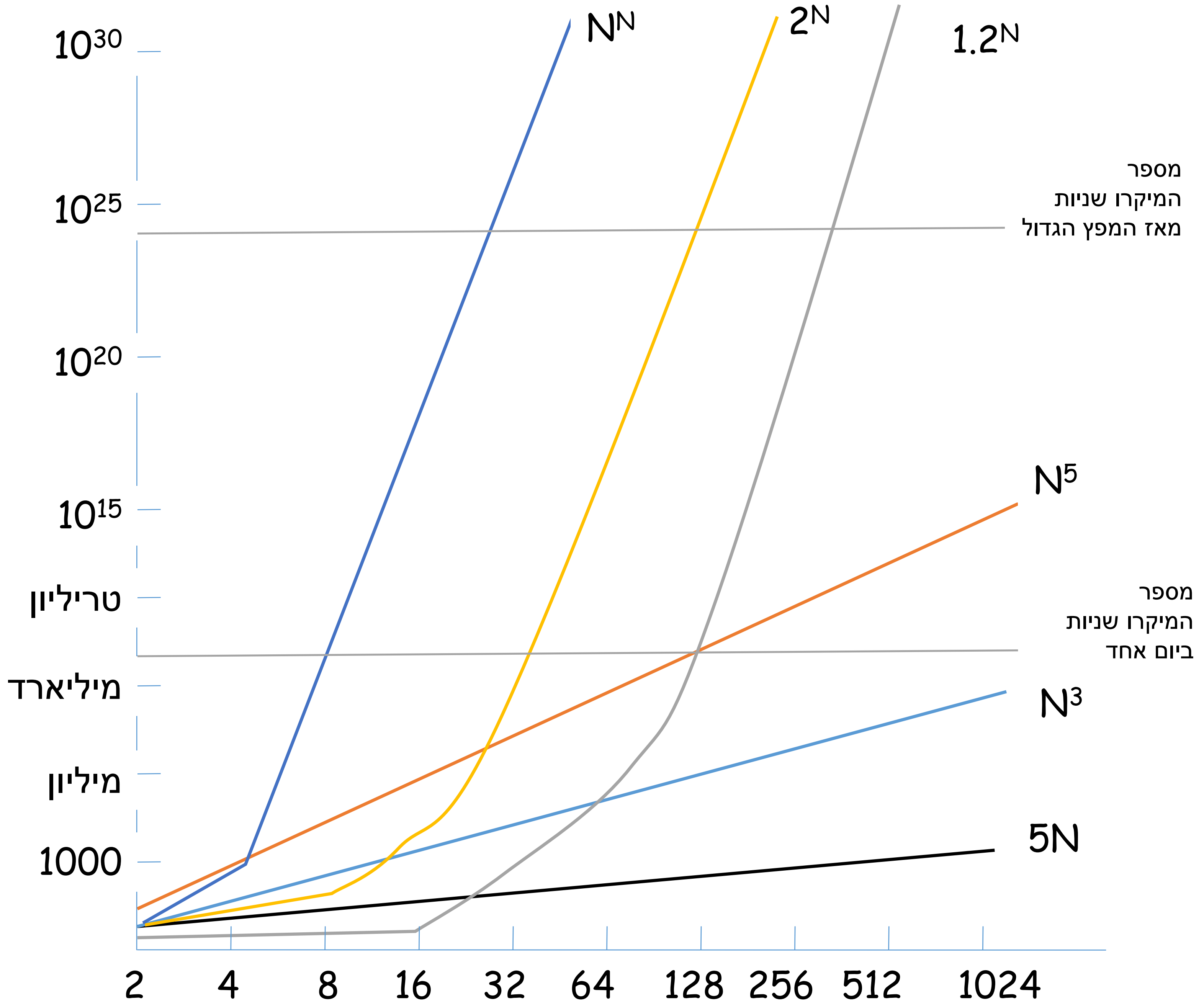
המטרה: מציאת יחס סדר על פונקציות.

הבעיה: פונקציה איננה מספר.

איך נשווה בין גודלי פונקציות?


מספר
צעדי
ריצה/
שניות/
מיקרו-
שניות
10^{30}
10^{25}
10^{20}
10^{15}
טריליון
מיליארד
מיליון
1000
2
4
8
16
32
64
128
256
512
1024
גודל
קלט


N^N
2^N
1.2^N
N^5
N^3
$5N$
10^{30}
10^{25}
10^{20}
10^{15}
טריליון
מיליארד
מיליון
1000
2
4
8
16
32
64
128
256
512
1024



N^N
2^N
1.2^N
10^{30}
10^{25}
מספר
המיקרו שניות
מאז המפץ הגדול
10^{20}
10^{15}
N^5
טריליון
מספר
המיקרו שניות
ביום אחד
מיליארד
N^3
מיליון
1000
5N
2
4
8
16
32
64
128
256
512
1024

$N \times \log_2 N$	N^2	N
33	100	10
665	10000	100
9966	מיליון	1000
20 מיליון	אלף מיליארדים	מיליון
30 מיליארדים	מיליארד מיליארדים	מיליארד

הפונ' / N	10	50	100	300	1000
5N	50	250	500	1500	5000
N×logN	33	282	665	2469	9966
N^2	100	2500	10000	90000	מיליון (7 ספרות)
N^3	1000	125000	מיליון (7 ספרות)	28 מיליון (8 ספרות)	מיליארד (10 ספרות)
2^N	1024	מספר בן 16 ספרות	מספר בן 31 ספרות	מספר בן 91 ספרות	מספר בן 302 ספרות
N!	3.6 מיליון (7 ספרות)	מספר בן 65 ספרות	מספר בן 161 ספרות	מספר בן 623 ספרות	מספר גדול לאין שיעור
N^N	10 מיליארד (11 ספרות)	מספר בן 85 ספרות	מספר בן 201 ספרות	מספר בן 774 ספרות	מספר גדול לאין שיעור

לצורך ההשוואה:

מספר הפרוטונים ביקום המוכר לנו הוא כבן 80 ספרות.

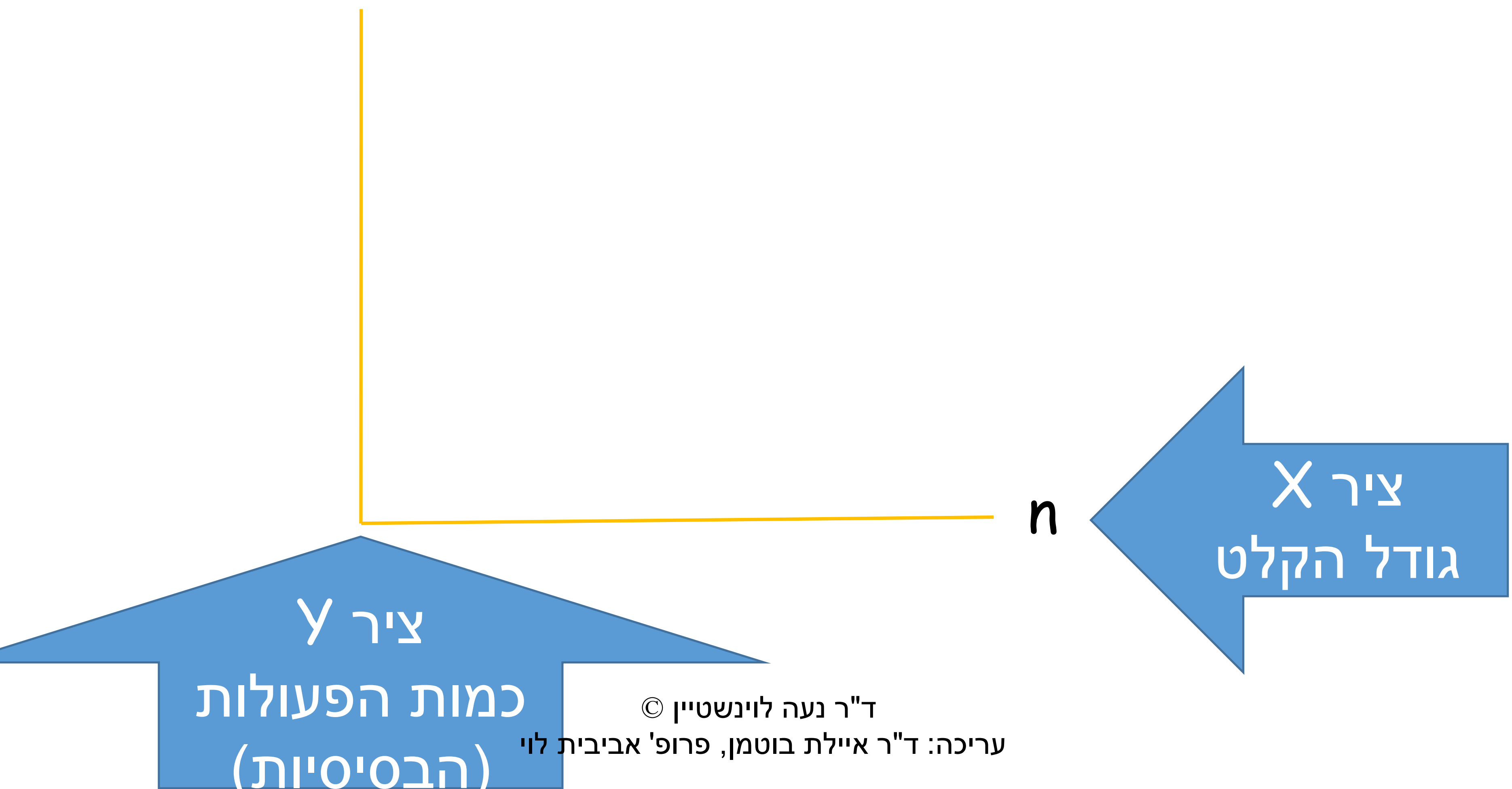
מספר המיקרו שניות שחלפו מאז המפץ הגדול הוא כבן 24 ספרות.

יעילות של אלגוריתמים סיבוכיות

<u>שיעור הגידול</u> - באופן פורמאלי

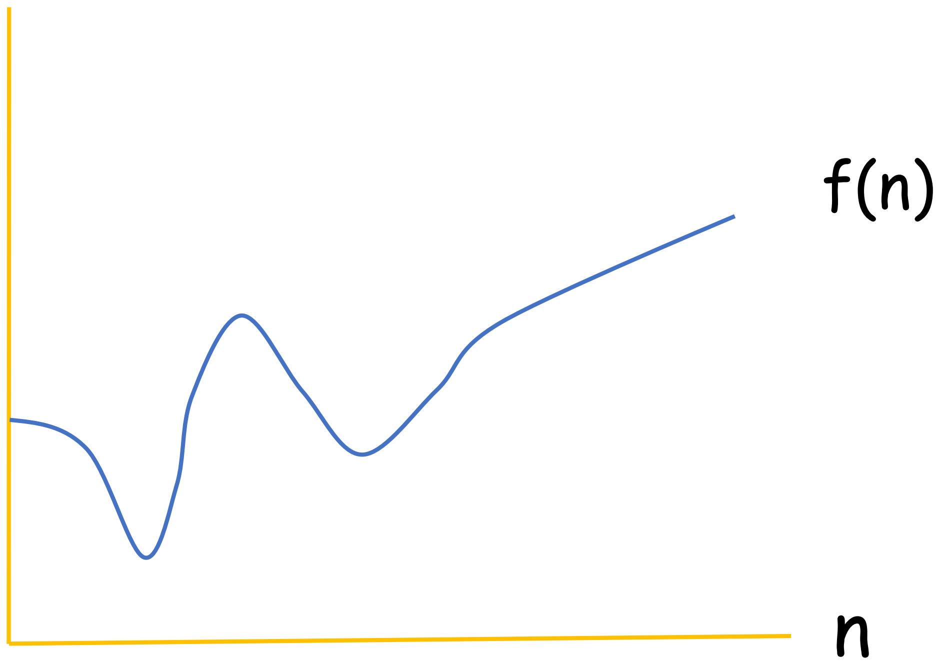
סימונים אסימפטוטיים

הסימון O
חסם אסימפטוטי עליון



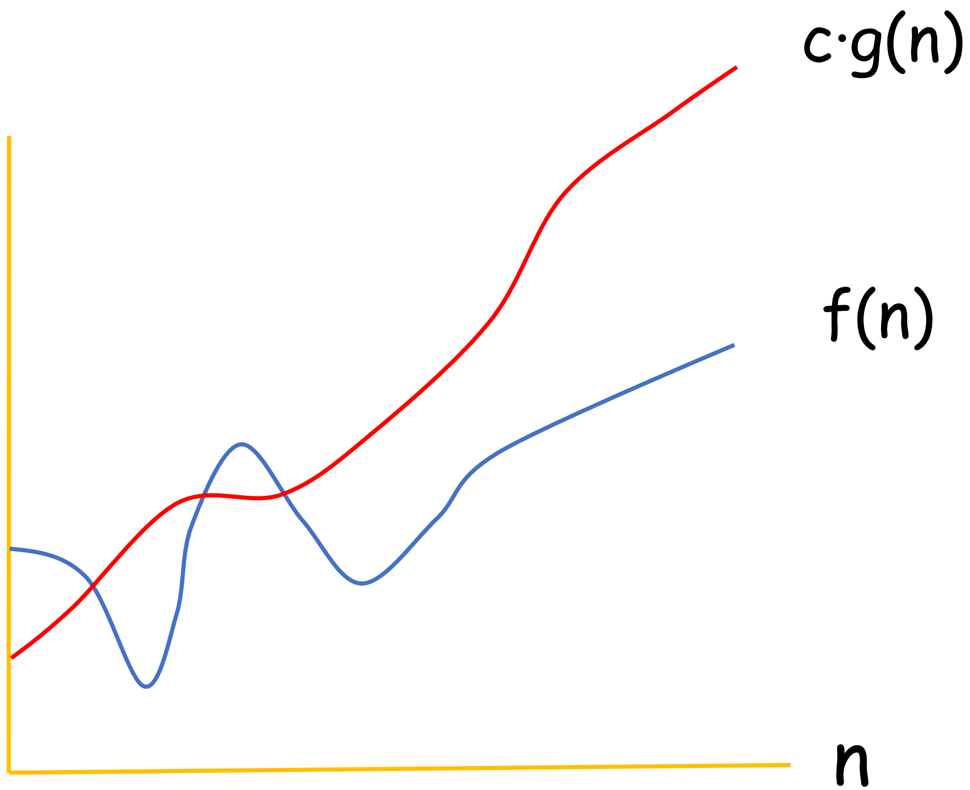
עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון O
חסם אסימפטומטי עליון

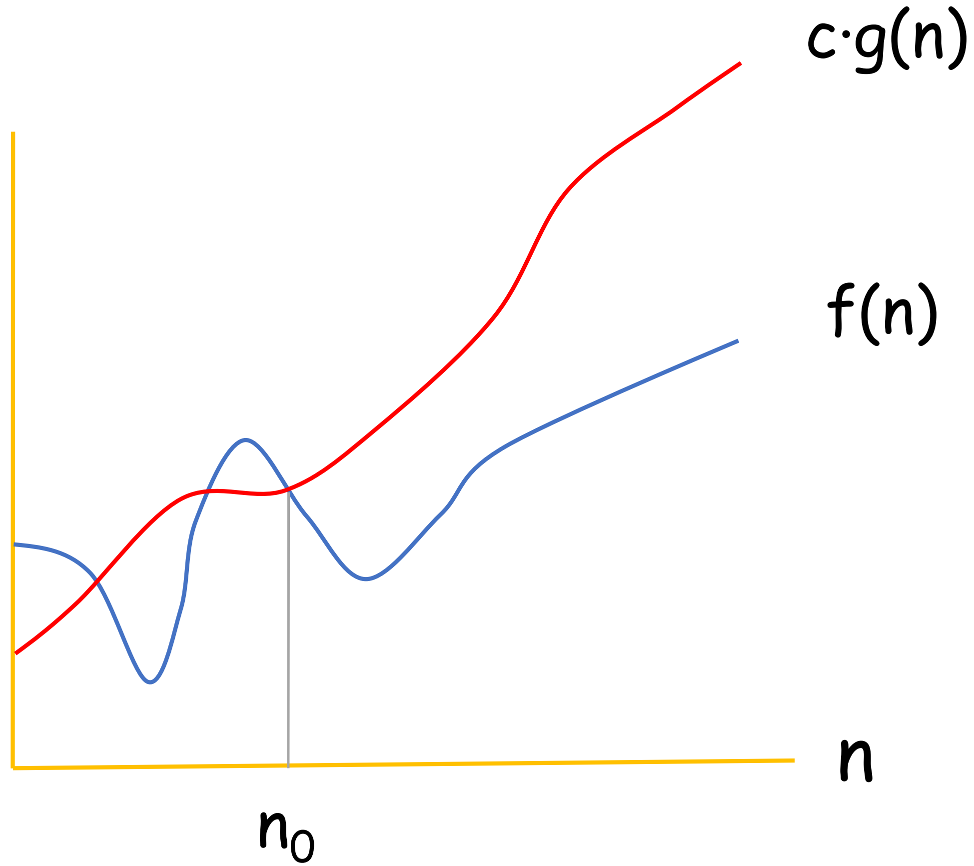


עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון O
חסם אסימפטומטי עליון

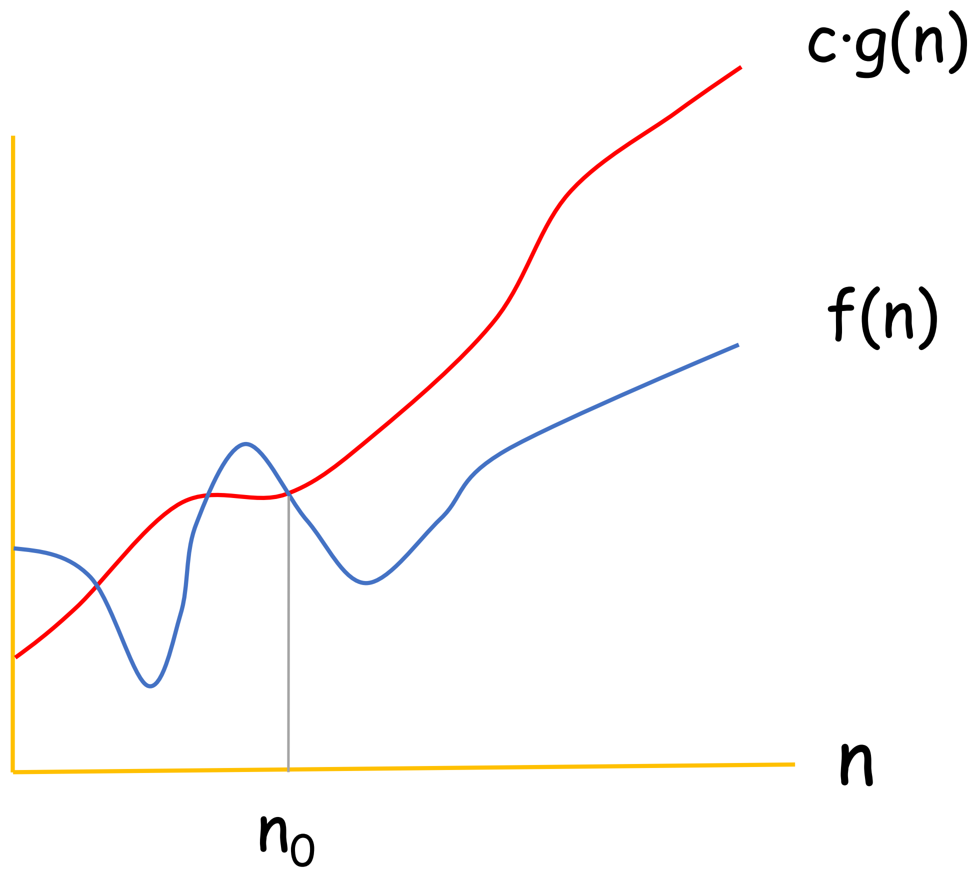


הסימון O
חסם אסימפטומטי עליון



עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון O
חסם אסימפטומטי עליון



$$f(n)=\boldsymbol{O}(g(n))$$

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון O

חסם אסימפטומטי עליון

$O(g(n))=$

$\{f(n) \mid$ קיימים קבועים חיוביים c ו- n_0 כך שלכל $n \geq n_0$ מתקיים $f(n) \leq c \cdot g(n)\}$

אם $f(n) \in O(g(n))$

נאמר ש-$g(n)$ הוא **חסם אסימפטומטי עליון** לפונקציה $f(n)$

ונסמן $f(n)=O(g(n))$

$O(g(n))=$

{קיימים קבועים חיוביים c ו- n_0 כך שלכל $n \geq n_0$ מתקיים $f(n) \mid f(n) \leq c \cdot g(n)$}

דוגמאות:

1. $3n^2+5 = O(n^2)$ נבחר: $c=4$ ו- $n_0=3$
2. $2n+5 = O(n^2)$ נבחר: $c=1$ ו- $n_0=5$
3. $2^{n+1} = O(2^n)$ נבחר: $c=4$ ו- $n_0=1$
4. $10n^{10} = O(2^n)$ נבחר: $c=10$ ו- $n_0=100$

$O(g(n))=$

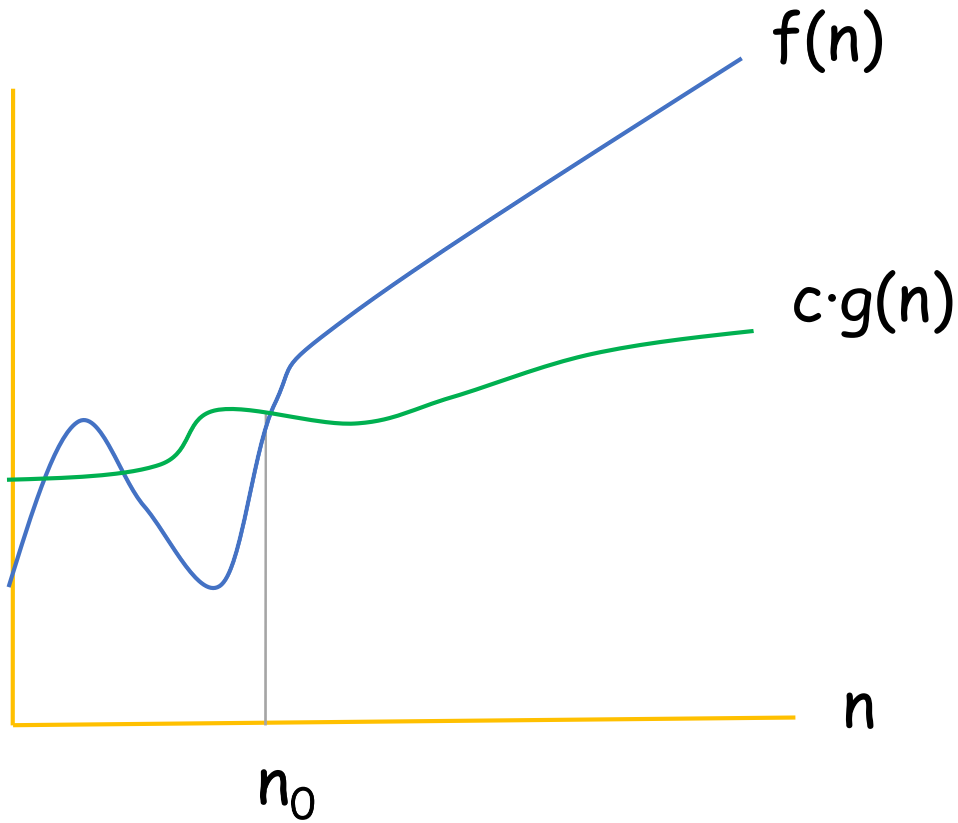
$\{f(n) \mid$ קיימים קבועים חיוביים c ו- n_0 כך שלכל $n \geq n_0$ מתקיים $f(n) \leq c \cdot g(n)\}$

דוגמאות:

1. $3n^2+5 = O(n^2)$ נבחר: $c=4$ ו- $n_0=3$
2. $2n+5 = O(n^2)$ נבחר: $c=1$ ו- $n_0=5$
3. $2^{n+1} = O(2^n)$ נבחר: $c=4$ ו- $n_0=1$
4. $10n^{10} = O(2^n)$ נבחר: $c=10$ ו- $n_0=100$

$2^{100} \sim 10^{30}$
$100^{10} = 10^{20}$

הסימון Ω
חסם אסימפטוטי תחתון



$$f(n) = \Omega(g(n))$$

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון Ω
חסם אסימפטומטי תחתון

$\Omega(g(n)) =$

{קיימים קבועים חיוביים c ו- n_0 כך שלכל $n \geq n_0$ מתקיים $f(n) \mid c \cdot g(n) \leq f(n)$}

אם $f(n) \in \Omega(g(n))$

נאמר ש-$g(n)$ הוא **חסם אסימפטומטי תחתון** לפונקציה $f(n)$

ונסמן $f(n)=\Omega(g(n))$

$\Omega(g(n))$

{$f(n) \mid c \cdot g(n) \leq f(n)$ קיימים קבועים חיוביים c ו- n_0 כך שלכל $n \geq n_0$ מתקיים}

דוגמאות:

1. $\Omega(n) \;=\; n^2 - 2$ נבחר: $c=1$ ו- $n_0=3$

2. $\Omega(n^3) \;=\; 2n^3 - 5n + 2$ נבחר: $c=1$ ו- $n_0=3$

$\Omega(g(n))$

{$f(n) \mid c\cdot g(n)\le f(n)$ מתקיים $n\ge n_0$ כך שלכל n_0 ו- c קיימים קבועים חיוביים}

דוגמאות:

1. $\Omega(n) = n^2 - 2$ נבחר: $c=1$ ו- $n_0=3$

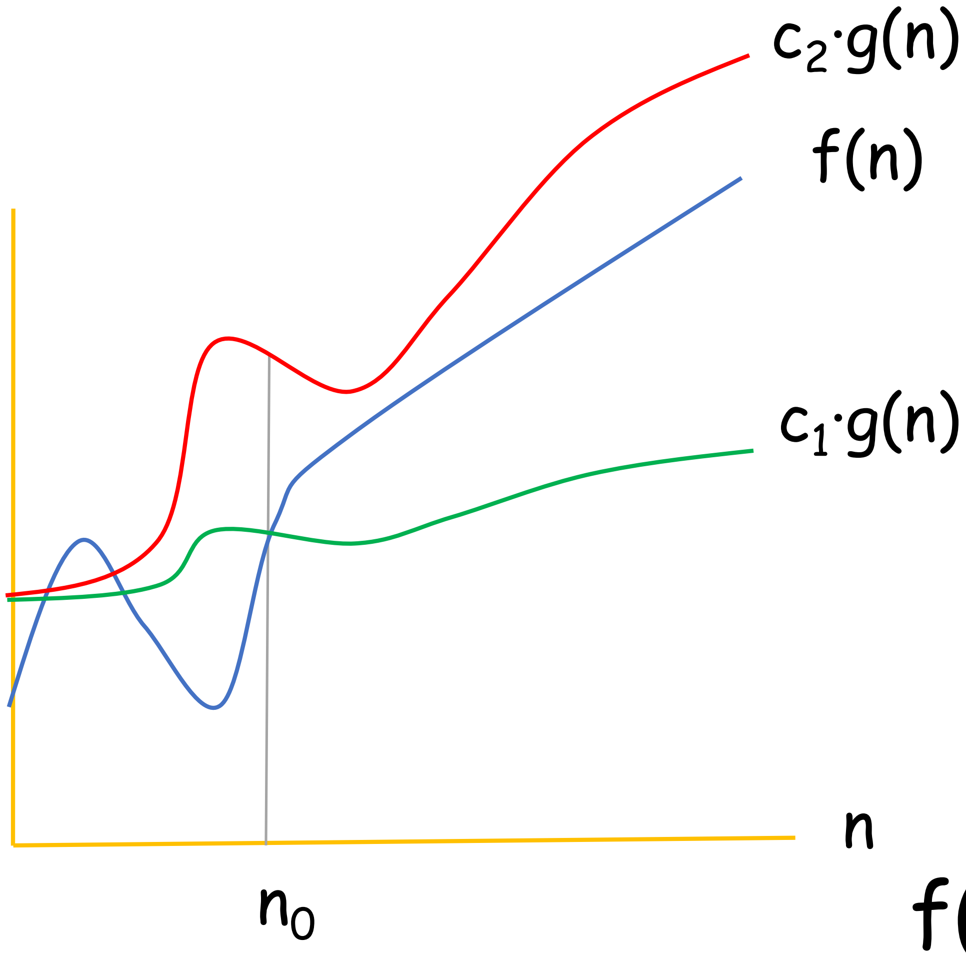
2. $\Omega(n^3) = 2n^3 - 5n + 2$ נבחר: $c=1$ ו- $n_0=3$

$$2n^3 - 5n + 2 < 2n^3$$

יש לבחור,
$0<c\le 1$

הסימון Θ

חסם אסימפטוטי הדוק



$$f(n)= \Theta(g(n))$$

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הסימון Θ
חסם אסימפטוטי הדוק

$\Theta(g(n)) =$

$\{f(n)|$

קיימים קבועים חיוביים c_1, c_2 ו- n_0 כך שלכל $n \geq n_0$ מתקיים $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)\}$

אם $f(n) \in \Theta(g(n))$

נאמר ש-$g(n)$ הוא **חסם אסימפטוטי הדוק** לפונקציה $f(n)$

ונסמן $f(n) = \Theta(g(n))$

דוגמא: $2n^3 - 5n + 2 = \Theta(n^3)$

עריכה: ד"ר איילת בוטמן, פרופ' אביבית לוי

הערה

1. לכל שתי פונקציות $f(n)$ ו- $g(n)$,

$f(n) = \Theta(g(n)) \iff f(n)=O(g(n))$ וגם $f(n)= \Omega(g(n))$

2. Θ הוא יחס שקילות

כלומר: רפלקסיבי - $f(n) = \Theta(f(n))$,

סימטרי - $f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n))$,

טרנסיטיבי - $f(n) = \Theta(g(n))$ ו $g(n) = \Theta(t(n)) \Rightarrow f(n) = \Theta(t(n))$.

תרגיל

האם מתקיים?

1. $2n^2 = O(n)$
2. $4^n = \Theta(2^n)$
3. $\log n = \Omega(n)$

$2n^2 = O(n)$

נניח בשלילה שקיימים קבועים c ו- n_0 כך ש- $2 \cdot n^2 \leq c \cdot n$ לכל $n_0 \leq n$.

נבחר $n > \max\{n_0, \frac{1}{2}c\}$ ונקבל שעבורו אי השוויון אינו מתקיים, כי $c \cdot n = c \cdot n \cdot \frac{1}{2} \cdot 2 < 2 \cdot n^2$.

סתירה !

$4^n = (2^2)^n = 2^{2n}$

$4^n = \Theta(2^n)$

נניח בשלילה שקיימים קבועים c ו- n_0 כך ש-
$2^{2n} \le c \cdot 2^n$ לכל $n_0 \le n$.
נוציא log:

$$\log(2^{2n}) \le \log(c \cdot 2^n)$$
$$2n \le \log c + n$$
$$n \le \log c$$

נבחר $n > \max\{n_0, c\}$ ונקבל שעבורו אי השוויון אינו מתקיים, כי $n > c > \log c$.

סתירה !

$\log n = \Omega(n)$

נניח בשלילה שקיימים קבועים c ו- n_0 כך ש-
$c \cdot n \le \log n$ לכל $n_0 \le n$.
ולכן:

$$1 \le \log n / c \cdot n$$

אבל מכיוון $\lim_{n \to \infty} \log n / c \cdot n = 0$

קיבלנו סתירה !

חסמים אסימפטומטיים

<u>דוגמאות נוספות</u>

- חלק למחלקות שקילות:

$5\log(n^2)$, $\log^2 n$, $5n^2$, $n!$, n^2-2n+5 , $\log n+2^n/10$, n^3-5n^2 , $(n+1)!$, 2^{2^n} , $10\cdot 2^n+n^2$

- סדר לפי סדר גודל עולה:

$\sqrt{n}$, $\log\log n$, $50\log^2 n$, $n^{3/2}/3$, 17 , $n\cdot\log n$, $7n$, 1.2^n , $17n\log n$, $7\log^7 n$, n^2+n

סיכום
הסימונים האסימפטומטיים

יהיו $f(n), g(n)\colon N \to R^+$ שתי פונקציות.

הגדרת O: נאמר ש- $g(n) = O(f(n))$, אם קיימים קבועים $c, n_0 > 0$ (c ממשי ו- n_0 טבעי) כך שלכל $n \geq n_0$ מתקיים $g(n) \leq c \cdot f(n)$.

הגדרת Ω: נאמר ש- $g(n) = \Omega(f(n))$, אם קיימים קבועים $c, n_0 > 0$ (c ממשי ו- n_0 טבעי) כך שלכל $n \geq n_0$ מתקיים $g(n) \geq c \cdot f(n)$.

הגדרת Θ: נאמר ש- $g(n) = \Theta(f(n))$, אם קיימים קבועים $c_1, c_2, n_0 > 0$ (c_1 ו- c_2 ממשיים ו- n_0 טבעי) כך שלכל $n \geq n_0$ מתקיים $c_1 f(n) \leq g(n) \leq c_2 f(n)$.

הגדרת o: נאמר ש- $g(n) = o(f(n))$, אם לכל קבוע $c>0$, קיים $n_0 > 0$ (c ממשי ו- n_0 טבעי) כך שלכל $n \geq n_0$ מתקיים $g(n) < c \cdot f(n)$.

הגדרת ω: נאמר ש- $g(n) = \omega(f(n))$, אם לכל קבוע $c>0$, קיים $n_0 > 0$ (c ממשי ו- n_0 טבעי) כך שלכל $n \geq n_0$ מתקיים $g(n) > c \cdot f(n)$.

הגדרות אלטרנטיביות עבור o קטן ו-ω קטן

הגדרת o: נאמר ש- $f(n) = o(g(n))$, אם $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

הגדרת ω: נאמר ש- $f(n) = \omega(g(n))$, אם $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

סימנים אסימפטומטיים - אינטואיציה

o is like $<$,

O is like $\leq$,

ω is like $>$,

Ω is like $\geq$,

Θ is like $=$.

סימנים אסימפטוטיים - הערה

לא כל שתי פונקציות ניתנות להשוואה

לדוגמא:

$T_1(n)=n$

$T_2(n)=n^{1+\sin n}$

טורים נפוצים בשימוש

- טור אריתמטי (סדרה חשבונית):

$$\sum_{k=1}^{n} k = 1+2+\ldots+n = \frac{1}{2}n(n+1)$$

- טור גיאומטרי (סדרה הנדסית):

$$\sum_{k=0}^{n} x^k = 1+x+x^2+\ldots+x^n = \frac{x^{n+1}-1}{x-1}$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \qquad \text{עבור} \quad |x|<1$$

- טור הרמוני:

$$H_n = 1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n} = \sum_{k=1}^{n}\frac{1}{k} = \ln n + O(1)$$

- סדרת הריבועים:

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$