

# רשימה מקושרת

Linked List



# מבוא

עד כה ביססנו את כל מבני הנתונים שלנו על מערכים.  
פתרון זה אינו תמיד יעיל.

0	1	2	3	4	5
11	22	13	-5	14	-4

1. גודל המערך נקבע עם היווצרו.

2. אם רוצים לשמור על סדר האיברים במערך, הכנסה  
והוצאה של איברים הן פעולות יקרות.

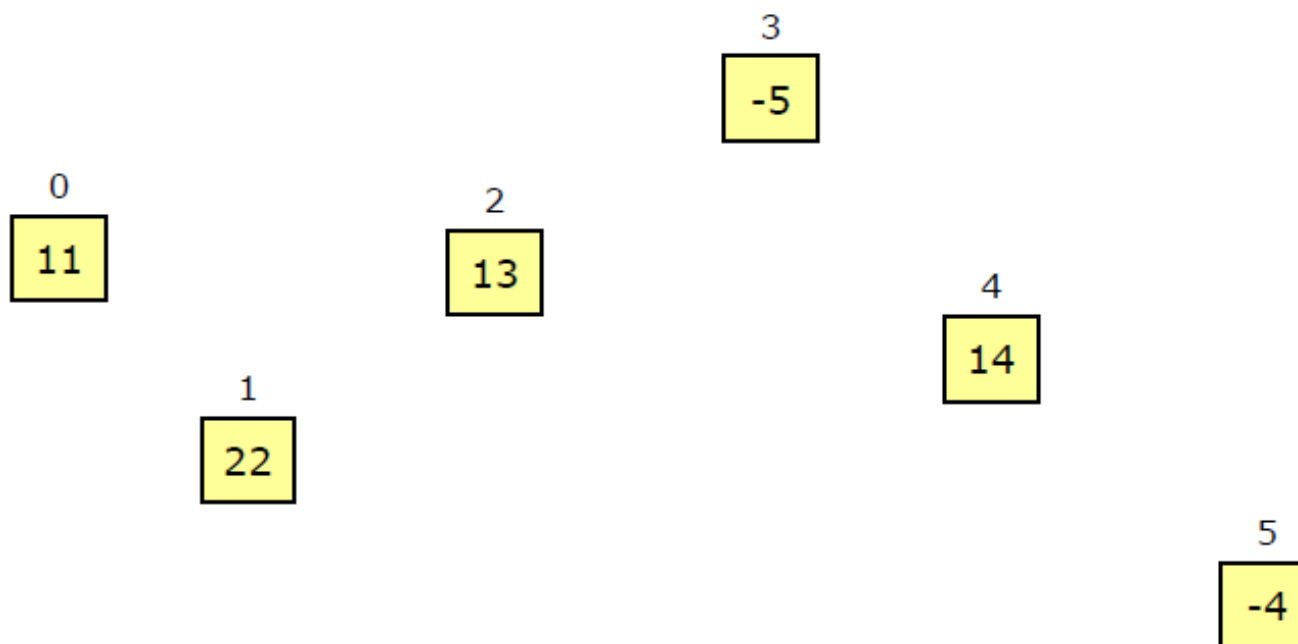
מערך - רצף של תאים (משתנים) זהים בזיכרון

**יתרון:** גישה ישירה.

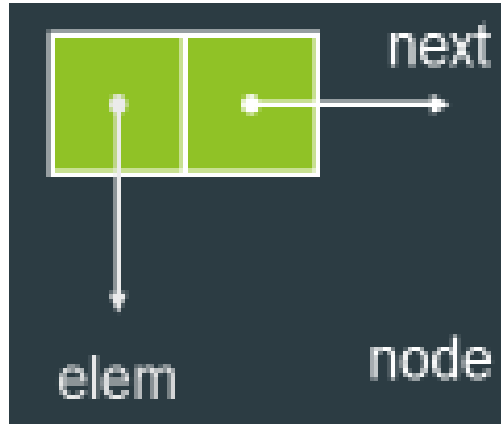
אם המשתנים מטיפוס `int` (4 בیتی) וכתובת המערך מתחילה ב-1000, אז כדי להגיע לתא עם אינדקס  $i$ ,  
המחשב עושה חישוב  $1000 + i * 4$  וכך מקבלים גישה ישירה.

# רשימה מקושרת

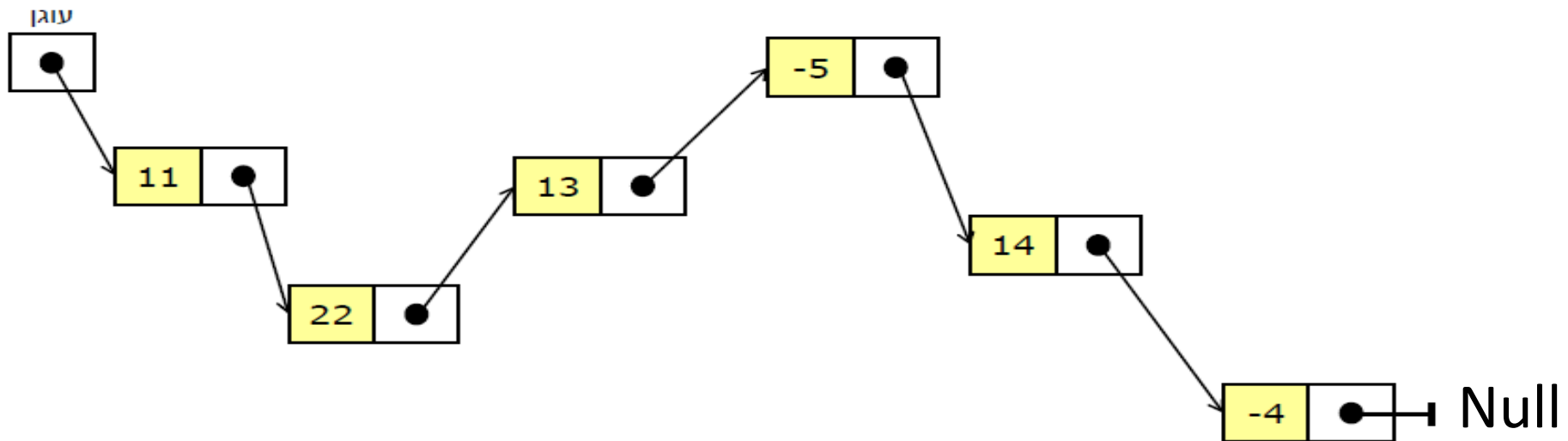
• נהפוך כל איבר לזוג: ערך + כתובת בזיכרון של האיבר הבא



# רשימה מקושרת

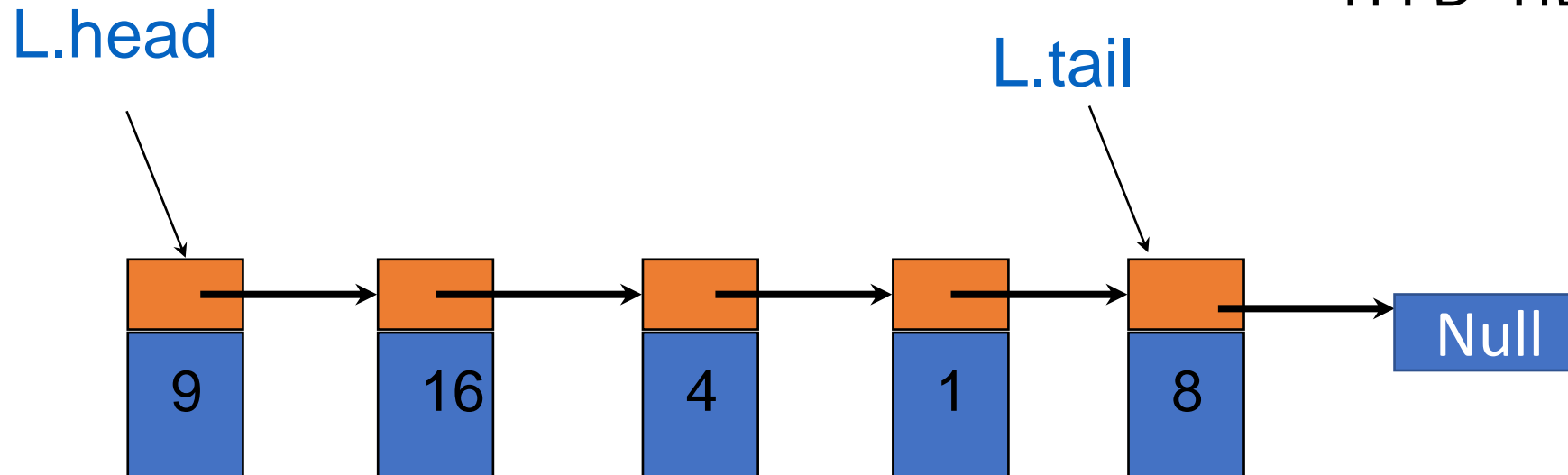


- רשימה מקושרת היא מבנה נתונים שבו האיברים מסודרים בסדר לינארי, הסדר נקבע ע"י מצביעים, כלומר כל איבר מצביע על איבר אחר.
- בד"כ המצביע נקרא next.



# רשימה מקושרת

- האיבר הראשון ברשימה נקרא ראש (head) והאחרון זנב (tail).
- עבור רשימה L, נשמר מאפיין L.head שמצביע על ראש הרשימה.
- לרוב נוסף מאפיין L.tail שיצביע על זנב הרשימה.
- יתרון – גמישות
- חסרון – אין גישה ישירה



# חיפוש ברשימה מקושרת

- הפעולה  $List - Search(L, k)$  מחזירה את המצביע לאיבר הראשון ברשימה בעל מפתח  $k$ . אם אין כזה הפעולה מחזירה Null.

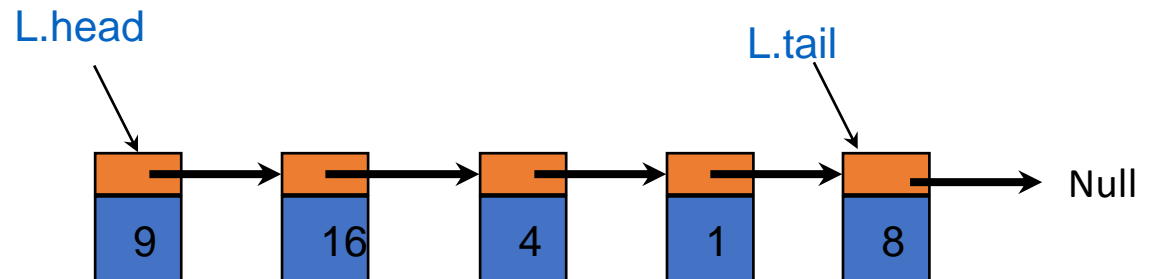
$List - Search(L, k)$

$x \leftarrow L.head$

$while (x \neq Null \text{ and } x.key \neq k)$

$x \leftarrow x.next$

$return x$



# שאלה

איזה איבר יוחזר בביצוע  $List - Search(L, 4)$  על הרשימה  $L$  הבאה?

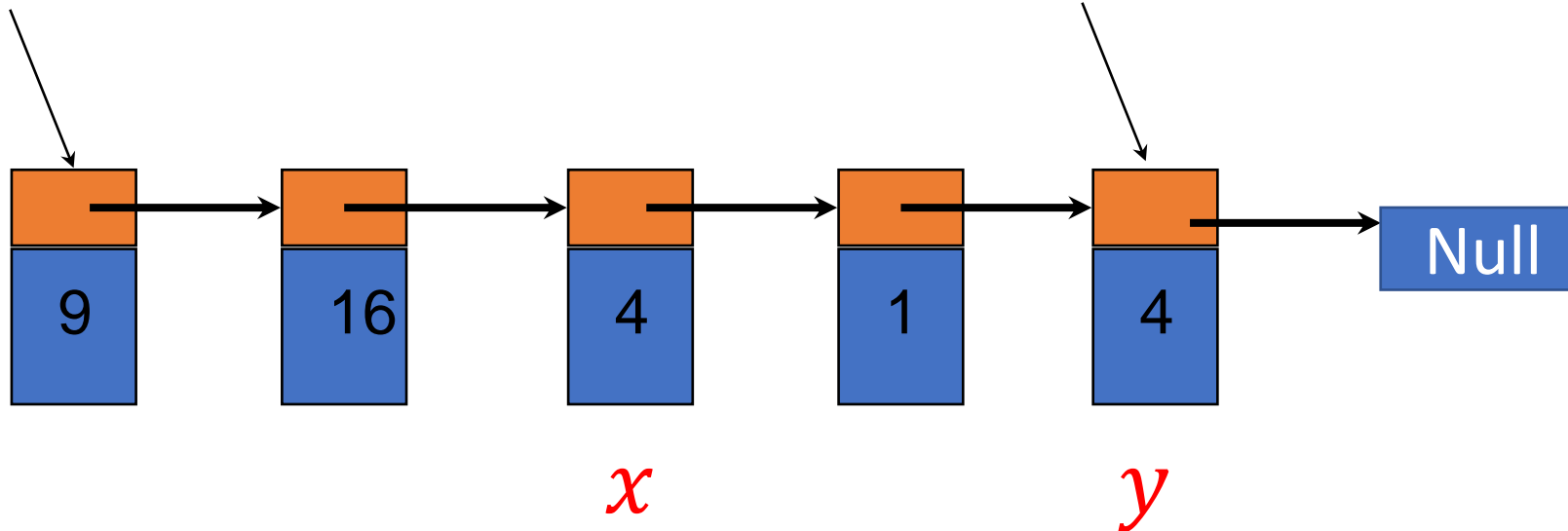
1.  $x$

2.  $y$

3. שניהם

L.head

L.tail



# חיפוש ברשימה מקושרת

מהי סיבוכיות פעולת החיפוש על רשימה בת  $n$  איברים במקרה הגרוע?

1.  $O(n)$

2.  $O(1)$

3.  $O(n^2)$

*List – Search( $L, k$ )*

*$x \leftarrow L.head$*

*while ( $x \neq \text{Null}$  and  $x.key \neq k$ )*

*$x \leftarrow x.next$*

*return  $x$*

Run time  $O(n)$  , in WS it may have to search the entire list



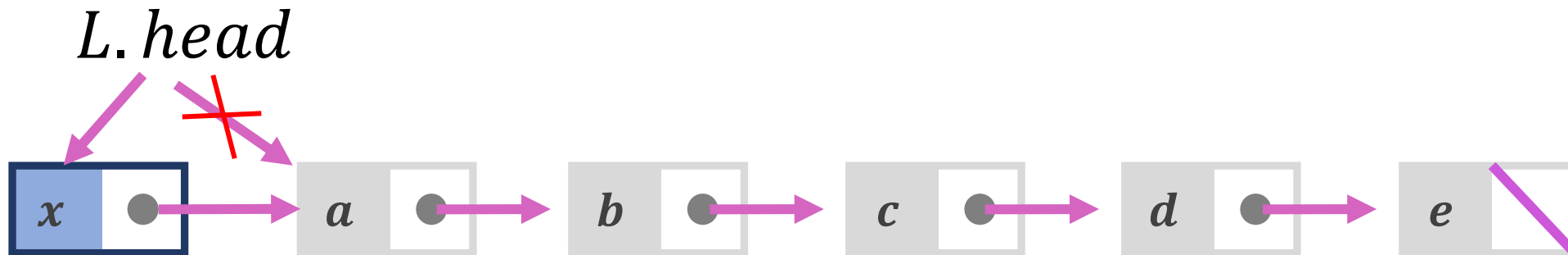
# הכנסה לרשימה מקושרת

• הפעולה  $List - Insert(L, x)$  מכניסה איבר חדש,  $x$ , לראש הרשימה.

$List - Insert - AtHead(L, x)$

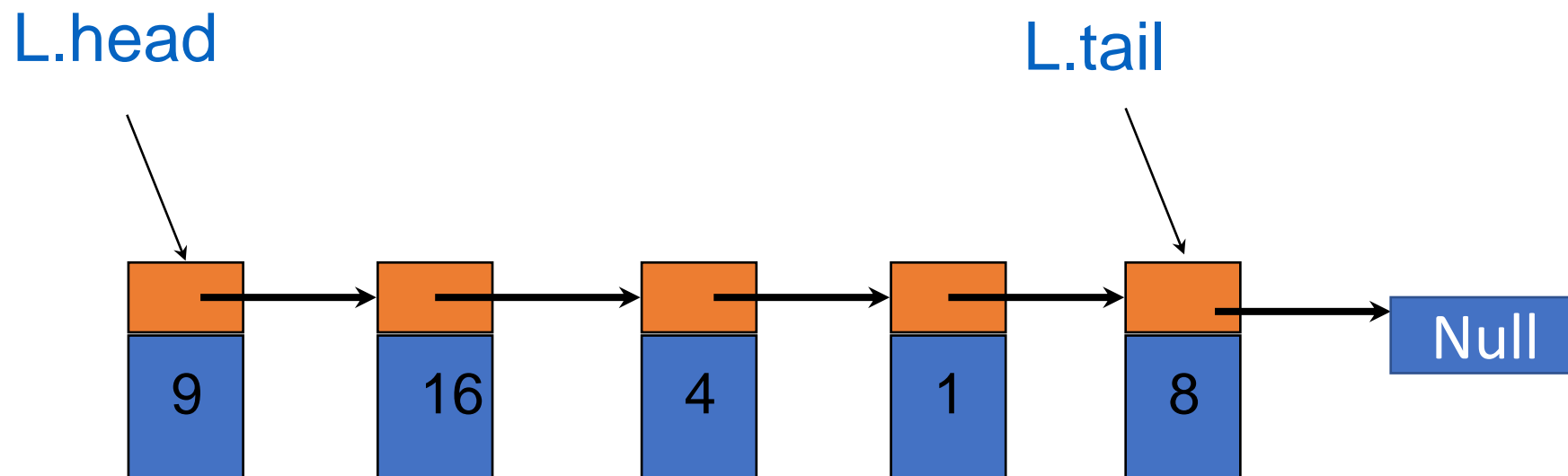
$x.next \leftarrow L.head$

$L.head \leftarrow x$



# הכנסה ברשימה מקושרת

- איך תראה הרשימה אחרי הוספת איבר עם המפתח 5 בראש הרשימה?

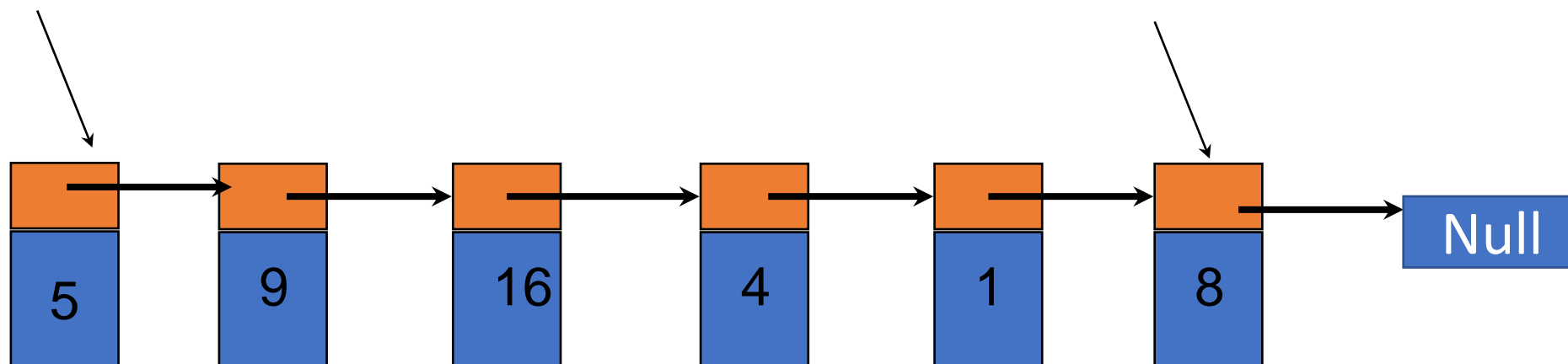


# הכנסה ברשימה מקושרת

- איך תראה הרשימה אחרי הוספת איבר עם המפתח 5 בראש הרשימה?

L.head

L.tail



# הכנסה לרשימה מקושרת

מהי סיבוכיות פעולת ההכנסה על רשימה בת  $n$  איברים במקרה הגרוע?

*List – Insert – AtHead( $L, x$ )*

*$x.next \leftarrow L.head$*

*$L.head \leftarrow x$*

.1  $O(n)$

.2  $O(1)$

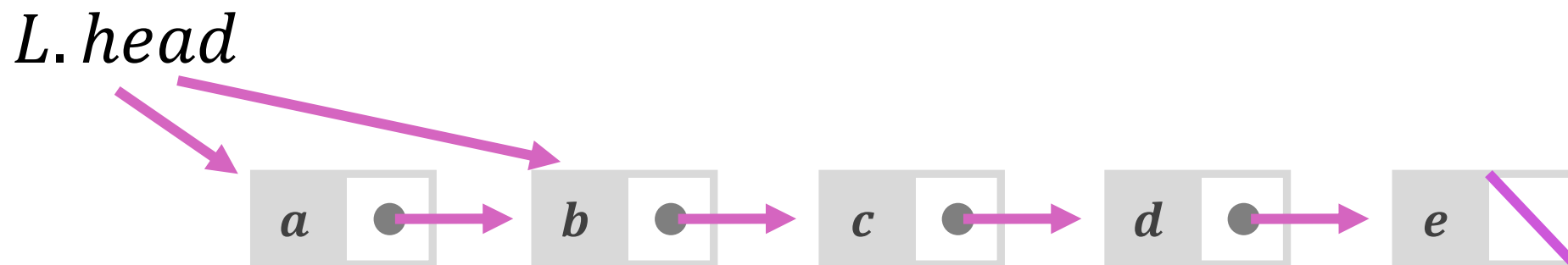
.3  $O(n^2)$

Running time  $O(1)$

# מחיקת הראש מרשימה מקושרת

- הפעולה  $List - Delete - Head(L)$  מוחקת את ראש הרשימה  $L$  ומחזירה מצביע לצומת שנמחק.

```
List - Delete - Head(L)  
   $x \leftarrow L.head$   
  if ( $L.head \neq Null$ )  
     $L.head \leftarrow (L.head).next$   
  return  $x$ 
```



# מחיקת הראש מרשימה מקושרת

מהי סיבוכיות פעולת המחיקה על רשימה בת  $n$  איברים במקרה הגרוע?

*List – Delete – Head(L)*

1.  $O(n)$

$x \leftarrow L.head$

2.  $O(1)$

*if* ( $L.head \neq Null$ )

3.  $O(n^2)$

$L.head \leftarrow (L.head).next$

return  $x$

Running time  $O(1)$

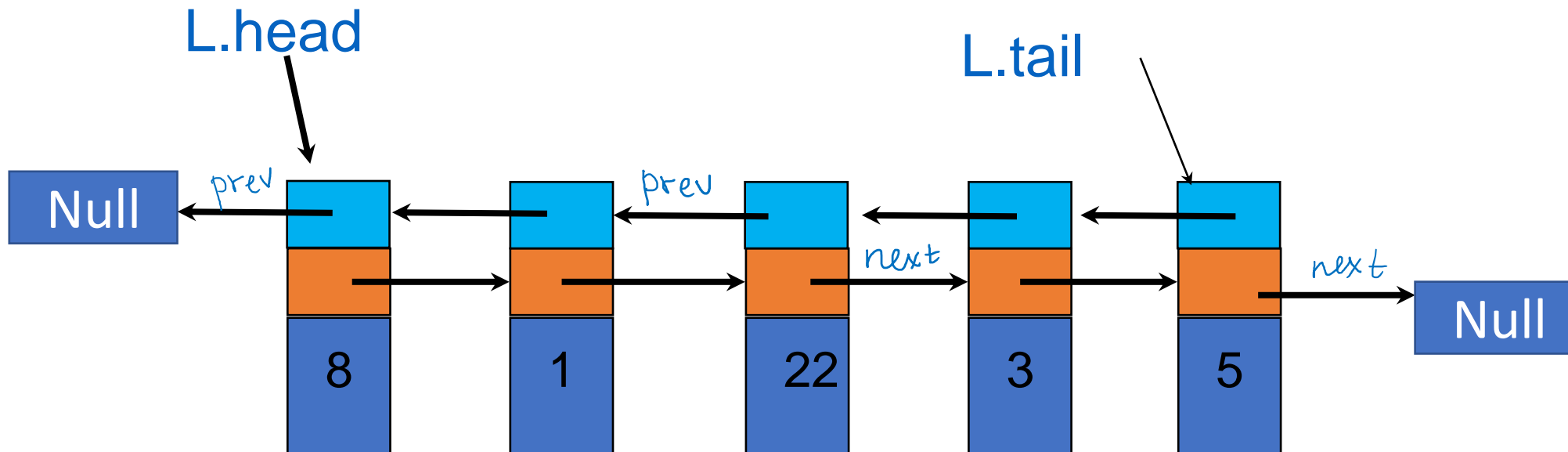
# מחיקה מרשימה מקושרת

- איך נממש מחיקת זנב הרשימה? מה הסיבוכיות?
- איך נממש מחיקת איבר בעל מפתח  $k$  מרשימה? מה הסיבוכיות?
- בתרגול ועבודת הבית...

Run time  $O(n)$ , we need to search  
previous element to update the pointers

# רשימה מקושרת דו-כיוונית

- רשימה דו מקושרת היא רשימה שבה כל איבר מצביע על הבא אחריו (מצביע next) וגם על קודמו (מצביע prev).
- מצביע prev של ראש הרשימה מצביע על Null.

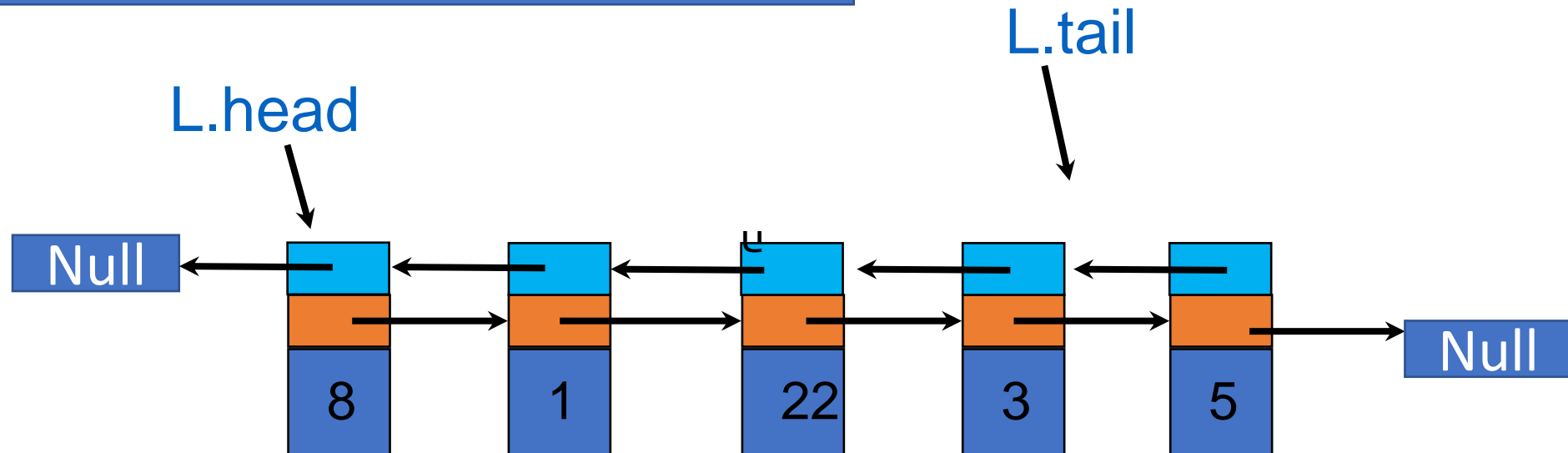




# הכנסה לרשימה מקושרת דו-כיוונית

```
List – Insert(L, x)
  x.next ← L.head
  if (L.head ≠ Null)
    (L.head).prev ← x
  L.head ← x
  x.prev ← Null
```

- הכנסת איבר בראש רשימה מקושרת דו-כיוונית.



# הכנסה לרשימה דו מקושרת

• מה סיבוכיות הפעולה על רשימה בת  $n$  איברים?

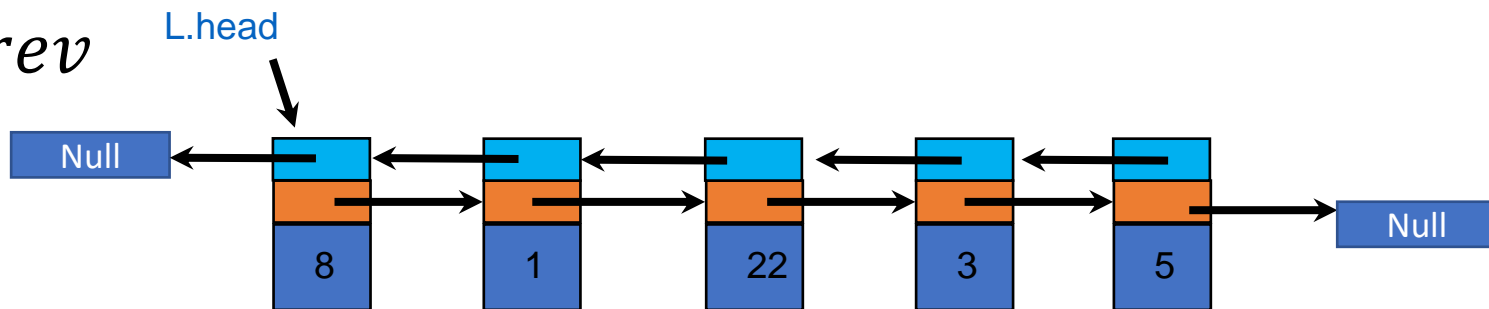
```
List – Insert( $L, x$ )  
   $x.next \leftarrow L.head$   
  if ( $L.head \neq \text{Null}$ )  
     $(L.head).prev \leftarrow x$   
   $L.head \leftarrow x$   
   $x.prev \leftarrow \text{Null}$ 
```

Running time  $O(1)$

# מחיקה מרשימה מקושרת דו-כיוונית

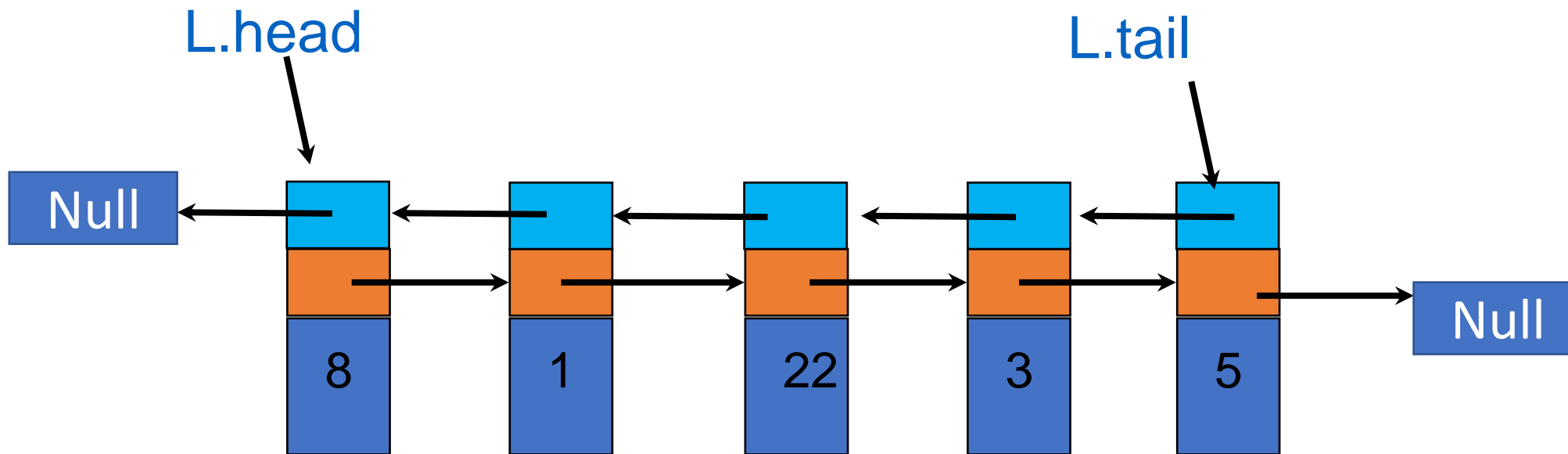
- הפעולה  $List - Delete(L, x)$  מוחקת את האיבר כלשהו  $x$  מהרשימה  $L$ . הפעולה מקבלת מצביע ל- $x$

```
List - Delete(L, x)
  if (x.prev ≠ Null)
    (x.prev).next ← x.next
  else
    L.head ← x.next
    (L.head).prev ← Null
  if (x.next ≠ Null)
    (x.next).prev ← x.prev
  else
    L.tail ← x.prev
    (L.tail).next ← Null
```



# מחיקה ברשימה דו מקושרת

- איך תראה הרשימה אחרי מחיקת האיבר 22?



# מחיקה מרשימה דו מקושרת

## שאלה 1:

מה סיבוכיות הפעולה  
*List – Delete* על רשימה בת  
 $n$  איברים כאשר נתון מצביע  
לצומת  $x$  שיש למחוק?

1.  $O(1)$

2.  $O(n)$

3.  $O(n^2)$

## שאלה 2:

במידע ונתון רק הערך (מפתח) של  
האיבר מה תהיה הסיבוכיות של הפעולה  
*List – Delete* ?

1.  $O(1)$

2.  $O(n)$

3.  $O(n^2)$

# סיכום

- הכרנו מבנה נתונים חדש – **רשימה**.
- רשימה היא אוסף לינארי של איברים שאינו מוגבל בגודלו והוא מאורגן כסדרה
- ניתן להכניס איברים לכל מקום ולהוציא איברים מכל מקום ברשימה