

נתונה המחלקה – Range – טווח, ולה שתי תכונות:

- low – מספר מטיפוס שלם
- high – מספר מטיפוס שלם

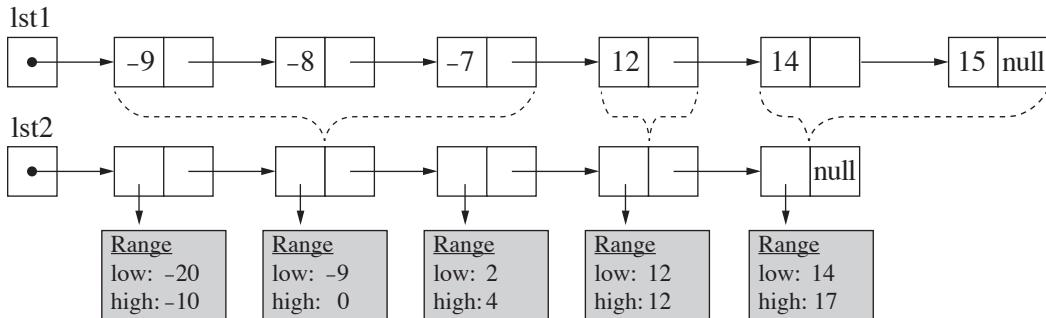
המספר high גדול או שווה ל- low ( $high \geq low$ ) .

הנראה שיש פועלות get/Set בubo תכונות המחלקה.

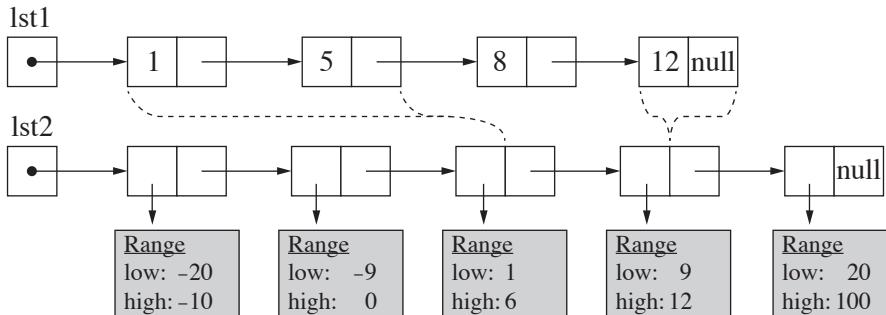
מספר כלשהו, x , ”מוכל” בעצם מטיפוס Range אם הוא נמצא בטווח המספרים שבין low ובין high (  $high \geq x \geq low$  ) .

שרשרת חוליות – lst1 מטיפוס שלם ”מוכלת” בשרשראת חוליות – lst2 מטיפוס Range אם בעבו בל מספר בשרשראת lst1 קיימת חוליה בשרשראת lst2 המכילה אותו.

דוגמה לשרשראת lst1 המוכלת בשרשראת lst2 :



דוגמה לשרשראת lst1 שאינה מוכלת בשרשראת lst2 :



הסביר: המספר 8 שברשראת lst1 אינו ”מוכל” בשום חוליה בשרשראת lst2 .

משמעות הפעולה החיצונית של להן:

**Java** – public static boolean isIncluded (Node<Integer> lst1, Node<Range> lst2)

**C#** – public static bool IsIncluded (Node<int> lst1, Node<Range> lst2)

הפעולהמחזירה true אם lst1 "מוכלת" ב- lst2, אחרת היא מחזירה false . הפעולה חייבת לעבוד בסיבוכיות זמן ריצה של  $O(N)$  .

הערה: N הוא אורך השרשרת הארוכה יותר מבין שתי השרשראות.

הנחות:

lst1 ו- lst2 אינם null –

בשערת lst2 כל העצמים מטיפוס Range אינם null –

**השערת lst1 ממויינת בסדר עולה.** –

**השערת lst2 ממויינת בסדר עולה,** כלומר, ערך ה- high של כל חוליה קטן מערך ה- low של החוליה הבאה אחריה בשערת (כפי ש�示 בדוגמאות בעמוד הקודם).