

נתונה המחלקה **Range** – טווח, ולה שתי תכונות:

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

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

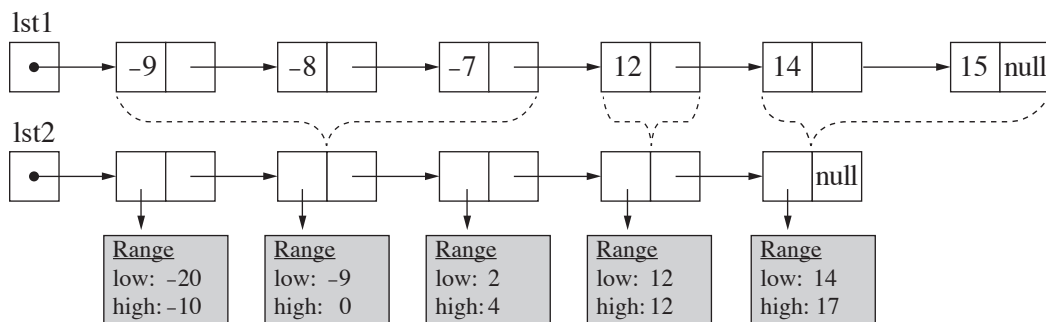
הניחו שיש פעולות get/Get ו- set/Set בעבור תכונות המחלקה.

מספר כלשהו, x, "מוכל" בעצם מטיפוס Range אם הוא נמצא בטווח המספרים שבין low ובין high

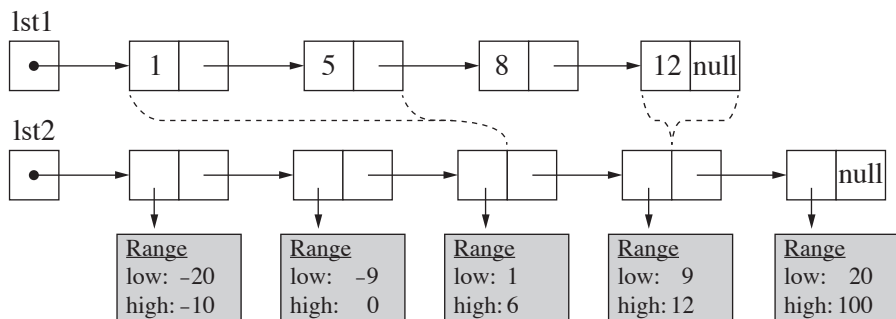
($high \geq x \geq low$).

שרשרת חוליות – lst1 מטיפוס שלם "מוכלת" בשרשרת חוליות – lst2 מטיפוס Range אם בעבור כל מספר בשרשרת lst1 קיימת חוליה בשרשרת lst2 המכילה אותו.

דוגמה לשרשרת lst1 המוכלת בשרשרת lst2 :



דוגמה לשרשרת lst1 שאינה מוכלת בשרשרת lst2 :



הסבר: המספר 8 שבשרשרת lst1 אינו "מוכל" בשום חוליה בשרשרת lst2.

ממשו את הפעולה החיצונית שלהלן:

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

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

הפעולה מחזירה 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.

– בשרשרת lst2 כל העצמים מטיפוס Range אינם null.

– השרשרת lst1 ממוינת בסדר עולה.

– השרשרת lst2 ממוינת בסדר עולה, כלומר, ערך ה-high של כל חוליה קטן מערך ה-low של החוליה הבאה

אחריה בשרשרת (כפי שמופיע בדוגמאות בעמוד הקודם).