מבני נתונים – תרגיל 2 – מעשי

תאריך פרסום: 31.3.19
תאריך הגשה: 28.4.19
מרצה ומתרגלים אחראים: פרופסור פז כרמי, מור ביטון, גליה צימרמן.

מבנה העבודה

חלק א' + ב': מימוש המבנה ב-Java
חלק ג': חלק תיאורטי- ניתוח זמן הריצה של חלק מהשיטות.

הערות חשובות למימוש והגשה:

- מומלץ מאוד לקרוא את העבודה מההתחלה עד הסוף לפחות פעמיים לפני שתתחילו לפתור אותה.
- אין להשתמש במבני נתונים גנריים הקיימים ב - Java במימוש העבודה (מלבד מערכים בהם אפשר להשתמש). כלומר, עליכם לממש כל מבנה שתרצו בעצמכם. (במחלקת הבדיקה שלכם אתם רשאים להשתמש במבנים מוכנים של Java, כיוון שאתם לא מגישים את המחלקה הזאת).
- ניתן תמיד להניח שהקלט יהיה תקין. לא תתבקשו להוסיף מפתח שכבר קיים; לא תתבקשו להוסיף מספר מפתחות העולה על מספר המקסימום של המפתחות שקיבלתם בבנאי; לא תתבקשו למחוק מפתחות שלא קיימים במבנה; לא תקבלו מתכנית הבדיקה שלנו ערך null כפרמטר לאף אחת מהשיטות של הממשק.
- אין לשנות את הממשק DynamicSet.
- אתם יכולים להוסיף למחלקות FloorsArrayLink, FloorsArrayList שיטות כרצונכם. אנו נבדוק את השיטות שהוגדרו במסמך זה ואת זמן הריצה שלהן.
- ככלל עליכם לממש את השיטות, כך שזמן הריצה שלהן יהיה קטן ככל האפשר.
- **חשוב: כל קבצי הקוד של העבודה שאתם מגישים צריכים להיות ב- default package** (זה שנוצר אוטומטית ביצירת פרויקט חדש בeclipse). אין ליצור package חדש. package חדש ימנע העלאת הקבצים ל- submission system.
- עליכם להגיש קובץ מסוג zip בשם assignment2.zip המכיל בתוכו:
  - תיקיית src ובה קבצי הג'אווה של העבודה: FloorsArrayLink.java, FloorsArrayList.java, DynamicSet.java
  - מסמך PDF ששמו partC.pdf שבו תשובותיכם לשאלות של חלק ג'.
- העבודות יבדקו במעבדה, כלומר סביבת העבודה בה תיבדקנה העבודות הינה ג'אווה 8.
- עליכם לדאוג כי עבודותיכם יתקמפלו וירוצו בסביבת eclipse תחת גרסת Java הנזכרת לעיל.
- עבודות שלא יתקמפלו – יקבלו ציון 0.
- עבודותיכם יבדקו באמצעות כלי בדיקה אוטומטים הבודקים קורלציה בין עבודות. אין להעתיק! להזכירכם, המחלקה רואה בחומרה רבה העתקות.
- נרצה לראות קוד מתועד, מתוכנן היטב ויעיל שמייצג הבנה.

תיאור המבנה - FloorsArrayList

במשימה זו תממשו מבנה נתונים השומר קבוצת מספרים ממשיים (מפתחות) ב*רשימת מערך קומות*. מבנה הנתונים תומך ב- dynamic set ADT, זאת אומרת בפעולות חיפוש, הכנסה, הוצאה, עוקב, קודם, מינימום ומקסימום. ברשימת מערך קומות חלק מהפעולות מתבצעות ביעילות רבה יותר מאשר ברשימה רגילה (עם תלות בפרמטרים מסויימים שאינם מעיניינו כדי לממש את המבנה). להלן הגדרת המבנה.

על רשימת מערך הקומות לקיים את התנאים הבאים:

- רשימת מערך הקומות ממויינת בסדר עולה לפי המפתח.
- כל המפתחות באוסף שונים זה מזה.
- כל חוליה שומרת מפתח (מספר ממשי).
- המפתח הראשון ברשימה הינו מינוס אינסוף ואינו חלק מאיברי האוסף.
- המפתח האחרון ברשימה הינו אינסוף ואינו חלק מאיברי האוסף.
- לכל חוליה יש מערך של מצביעים קדימה ומערך של מצביעים אחורה (פירוט על תוכן המערכים יינתן בנקודות הבאות). שני המערכים בעלי גודל זהה.
- כל חוליה שומרת ערך $arrSize$ המציין את הגודל של מערכי המצביעים של החוליה. $arrSize$הינו מספר שלם הגדול מאפס.
- למפתח הראשון (מינוס אינסוף) ולמפתח האחרון (אינסוף) ערך זה ($arrSize$) גדול יותר ממספר האיברים ברשימה.
- לחוליה עם מפתח k שגודל המערך שלה הוא $arrSize$יש מערך של <u>מצביעים קדימה</u> בגודל $arrSize$. האיבר ה-$i+1$ במערך (אינדקס i, $0 \leq i < arrSize$) הוא מצביע לחולייה הקרובה ביותר <u>מימין</u> שגודל המערך שלה <u>גדול</u> (ממש) מ- i (ובמילים אחרות, מצביע לחוליה עם המפתח הקטן ביותר שגדול מ-k וגודל המערך של החוליה גדול מ-i).

  למשל באיור למטה, לחוליה עם המפתח 3 יש במקום הראשון (אינדקס $i = 0$) במערך המצביעים קדימה מצביע אל החוליה עם מפתח 4 כיוון שגודל המערך שלה (של החוליה עם מפתח 4) הוא 3 שגדול ממש מ-0. גם במקום השני (אינדקס $i = 1$) במערך המצביעים קדימה יש מצביע לחוליה עם מפתח 4 (גודל המערך 3 גדול ממש מ-1)

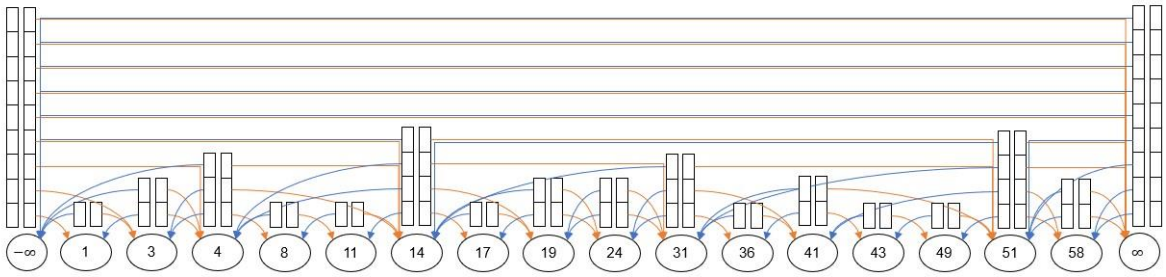
- לחוליה עם מפתח k שגודל המערך שלה הוא $arrSize$ יש מערך של <u>מצביעים אחורה</u> בגודל $arrSize$. האיבר ה-$i+1$ במערך (אינדקס i, $0 \leq i < arrSize$) הוא מצביע לחולייה הקרובה ביותר <u>משמאל</u> שגודל המערך שלה <u>גדול</u> (ממש) מ- i (ובמילים אחרות, מצביע לחוליה עם המפתח הגדול ביותר שקטן מ-k וגודל המערך של החוליה גדול מ-i).

  למשל באיור למטה, לחוליה עם המפתח 3 יש במערך המצביעים אחורה במקום הראשון (אינדקס $i = 0$) מצביע אל החוליה עם מפתח 1 שגודל המערך שלה הוא 1 ($1 > i$). במערך המצביעים אחרוה במקום השני (אינדקס $i = 1$) מצביע לחוליה עם מפחת $-\infty$, כיוון שגודל המערך של החוליה עם מפתח 1 הוא 1 (כלומר קטן מידי), בעוד גודל המערך של החוליה עם מפתח $-\infty$ גדול מ-1.

- עבור מבנה ריק של מפתחות, הרשימה תחזיק שתי חוליות: אחת עם מפתח אינסוף והשניה עם מפתח מינוס אינסוף. גודל האוסף הוא 0.

דוגמה:

להלן איור של מבנה הנתונים: המספרים באיור הם המפתחות; האוסף מכיל 16 מפתחות. שימו לב, המערכים של המפתחות אינסוף ומינוס אינסוף במבנה שלכם צריכים להיות יותר גדולים מכפי שהם באיור. לנוחיותכם, המצביעים קדימה צבועים בכתום והמצביעים אחורה - בכחול.



בנספח א' תוכלו למצוא איורים המדגימים את השינויים במבנה בעקבות רצף פעולות הכנסה והוצאה.

חלק א: מימוש FloorsArrayLink:

על כל השיטות של המחלקה FloorsArrayLink לרוץ בזמן יעיל ככל האפשר.

הוספת שדות:

הוסיפו O(arrSize) שדות כראות עיניכם.

מימוש השיטות:

1. **FloorsArrayLink(double key, int arrSize)**
   <u>בנאי</u> המקבל מספר ממשי key (מפתח), ואת גודל מערכי המצביעים קדימה ואחורה (arrSize). הבנאי בונה חוליה בהתאם.
   <u>זמן ריצה:</u> O(arrSize)

<u>על כל השיטות הבאות לרוץ בזמן O(1).</u>

2. **double getKey()**
   שיטה המחזירה את המפתח של החוליה.
3. **FloorsArrayLink getNext(int i)**
   השיטה מחזירה את המצביע ה-i קדימה של החוליה הנוכחית, כלומר השיטה מחזירה את המצביע השמור במערך המצביעים קדימה באינדקס ה- $i-1$. למשל, עבור רשימת מערך הקומות שבדוגמה לעיל, כאשר $i=3$, החוליה של 31 תחזיר את החוליה של 51. כאשר $i=4$ הפעולה לא חוקית עבור החוליה של 31, השיטה תחזיר null.
4. **FloorsArrayLink getPrev(int i)**
   השיטה מחזירה את המצביע ה-i אחורה של החוליה הנוכחית, כלומר השיטה מחזירה את המצביע השמור במערך המצביעים אחורה באינדקס ה- $i-1$. למשל, עבור רשימת מערך הקומות שבדוגמה לעיל, כאשר $i=3$, החוליה של 31 תחזיר את החוליה של 14. כאשר $i=4$ הפעולה לא חוקית עבור החוליה של 31, השיטה תחזיר null.
5. **void setNext(int i, FloorsArrayLink next)**
   השיטה משנה את המצביע ה-i קדימה של החוליה הנוכחית, כלומר השיטה מחליפה את המצביע השמור במערך המצביעים קדימה באינדקס $i-1$ במצביע ל-$next$. למשל, עבור רשימת מערך הקומות שבדוגמה לעיל, כאשר $i=3$, המצביע השלישי קדימה של החוליה של 31 לא ימשיך להצביע לחוליה של 51, אלא על next. כאשר $i=4$ הפעולה לא חוקית עבור החוליה של 31, השיטה לא תעשה דבר.
6. **void setPrev(int i, FloorsArrayLink prev)**
   השיטה משנה את המצביע ה-i אחורה של החוליה הנוכחית, כלומר השיטה מחליפה את המצביע השמור במערך המצביעים אחורה באינדקס $i-1$ במצביע ל-$prev$. למשל, עבור רשימת מערך הקומות שבדוגמה לעיל, כאשר $i=3$, המצביע השלישי אחורה של החוליה של 31 לא ימשיך

להצביע לחוליה של 14, אלא על prev. כאשר $i = 4$ הפעולה לא חוקית עבור החוליה של 31, השיטה לא תעשה דבר.

7. **int getArrSize()**
השיטה מחזירה את הגודל של מערכי המצביעים.

חלק ב: מימוש FloorsArrayList:

על כל השיטות של המחלקה FloorsArrayList לרוץ בזמן יעיל ככל האפשר. הניקוד של סעיף זה מחולק לשני רכיבים: נכונות ויעילות. ניקוד על יעילות המבנה יקבע בהתאם לזמן שיקח למבנה שלכם לבצע סדרה של n הכנסות, מחיקות וחיפושים.

הוספת שדות:

הוסיפו O(1) שדות כראות עיניכם.

מימוש השיטות:

1. **FloorsArrayList (int N)**
<u>בנאי</u> המקבל חסם עליון על מספר האיברים ברשימה, ומייצר רשימה ריקה כפי שהיא מוגדרת לעיל.
<u>זמן ריצה:</u> O(N).

2. **int size()**
שיטה המחזירה את גודל הרשימה.
<u>זמן ריצה:</u> O(1).

3. **FloorsArrayLink lookup(double key)**
השיטה מחזירה את החוליה שהמפתח שלה שווה ל-key. אם לא קיימת חוליה כזו, השיטה מחזירה null.

4. **void insert(double key, int arrSize)**
שיטה זו מקבלת מפתח (key) וגודל מערך מצביעים (arrSize). השיטה מכניסה את המפתח לרשימה לפי סדר המיון. לחוליה החדשה יש ליצור מערכי מצביעים קדימה ואחורה בגודל arrSize (כל אחד). בנוסף השיטה מעדכנת את המצביעים ברשימה (ובחוליה החדשה) כך שהרשימה תשמור על תכונותיה.

5. **void remove(FloorsArrayLink toRemove)**
שיטה זו מקבלת חוליה ששייכת למבנה ומסירה אותו. יש לשמור על תכונות הרשימה לאחר המחיקה.

6. **double successor(FloorsArrayLink link)**
השיטה מחזירה את המפתח העוקב של link במבנה. אם לא קיים כזה, השיטה מחזירה מינוס אינסוף.
<u>זמן ריצה:</u> O(1).

7. **double predecessor(FloorsArrayLink link)**
השיטה מחזירה את המפתח הקודם של link במבנה. אם לא קיים כזה, השיטה מחזירה אינסוף.
<u>זמן ריצה:</u> O(1)

8. **double minimum()**
השיטה מחזירה את המפתח המינימלי במבנה. אם המבנה ריק, יוחזר אינסוף.
<u>זמן ריצה:</u> O(1)

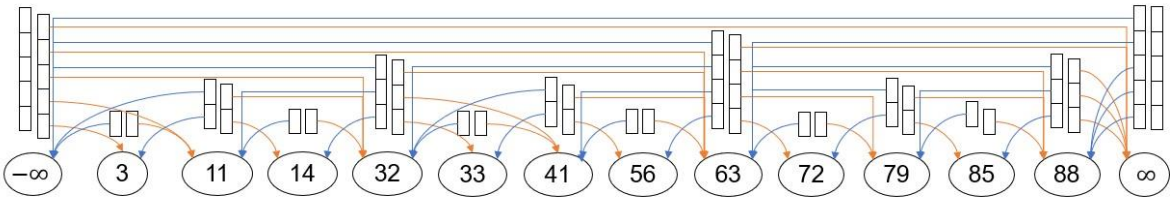
9. **double maximum()**
השיטה מחזירה את המפתח המקסימלי במבנה. אם המבנה ריק, יוחזר מינוס אינסוף.
<u>זמן ריצה:</u> O(1)

חלק ג: שאלות תאורתיות

את התשובות לחלק זה יש **<u>להקליד</u>** ולשמור בקובץ PDF בשם partC.pdf.

בחלק זה נניח כי המבנה בגודל n (מספר המפתחות הוא n) שבניתם מקיים תכונה נוספת:

- גודל המערך של החוליה ה- i $(1 \le i \le n)$ הינו $x+1$, כאשר x הינו המספר הטבעי הגדול ביותר שמקיים: $i \; mod \; 2^x = 0$ $(0 \le x \le \log n)$. כאשר n הוא מספר המפתחות ברשימת מערך הקומות.
  למשל, באיור למטה החוליה עם המפתח 14, הינה החוליה ה-3. ה-x הגדול ביותר המקיים את התנאי הינו 0 $(3 \; mod \; 2^0 = 0)$, ועל כן גודל המערכים הינו 1.
  למשל, החוליה השומרת את המפתח 63, הינה החוליה ה-8. ה-x הגדול ביותר המקיים את התנאי הינו 3 $(8 \; mod \; 2^3 = 0)$, ועל כן גודל המערכים הינו 4.
- שימו לב, החוליה ה-0, אשר לה מפתח מינוס אינסוף, והחוליה ה-$(n+1)$, אשר לה מפתח אינסוף, אינן עומדות בתנאי זה (כחלק מההגדרה). גודל המערך שלהן נשאר כמו קודם (גדול ממספר איברי המערך).
  באיור המערכים של חוליות אלה קטנים מכפי שהם צריכים להיות (אבל אין להן מצביעים נוספים ש**שונים** מאלה המוצגים באיור).



1. **חיפוש:**
   א. תארו את האלגוריתם לחיפוש מפתח ברשימה שכתבתם בסעיף הקודם (lookup).
   ב. נתחו את זמן הריצה של האלגוריתם שתיארתם בסעיף הקודם (במקרה הגרוע ביותר).
   זיכרו, עליכם לנתח את זמן הריצה במבנה שמקיים את התכונה שתוארה בתחילת חלק ג'.

2. **הכנסה:**
   נתחו את זמן הריצה <u>ההדוק ביותר</u> של אלגוריתם ההכנסה (insert) שכתבתם בסעיף הקודם (במקרה הגרוע ביותר). כלומר, הניחו שהמבנה עומד בכל תנאי המבנה (כולל התנאי החדש) ויש בו n מפתחות, והתבקשתם לבצע הכנסה יחידה- מה זמן הריצה של ההכנסה?

3. **הוצאה:**
   א. תארו את אלגוריתם ההוצאה שכתבת בסעיף הקודם (Remove).
   ב. נתחו את זמן הריצה <u>ההדוק ביותר</u> של האלגוריתם שתיארתם בסעיף הקודם (במקרה הגרוע ביותר). כלומר, הניחו שהמבנה עומד בכל תנאי המבנה (כולל התנאי החדש) ויש בו n מפתחות, והתבקשתם לבצע הוצאה יחידה- מה זמן הריצה של ההוצאה?

4. **גודל המבנה:**
   חיסמו מלמעלה את המקום שהמבנה דורש במידה והוא מכיל n מפתחות. הניחו שכל מספר וכל מצביע נשמרים ב $O(1)$. כך למשל, מערך של p מצביעים נשמר ב $O(p)$. הסבירו בפירוט את החישוב שלכם.
   עליכם לנתח את סיבוכיות המקום של מבנה המקיים את התכונה שתוארה בתחילת חלק ג'.

בהצלחה!

נספח א' – דוגמה לשינויים במבנה בעקבות הכנסות והוצאות

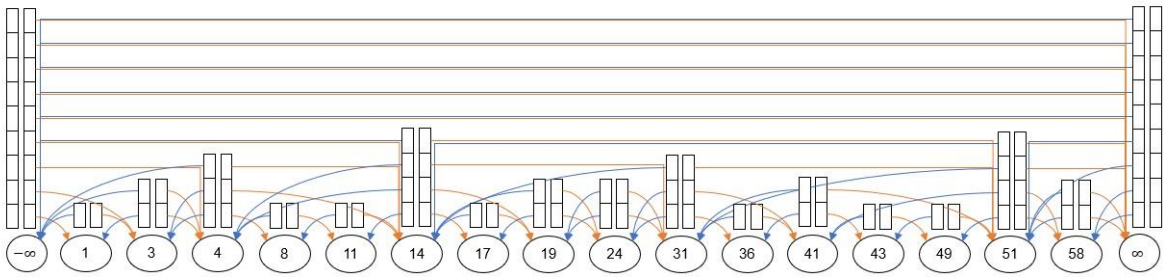
שימו לב, השינויים ברשימת מערך הקומות בעקבות כל פעולה נפרשו על פני מספר איורים על מנת שיהיה קל יותר להבחין בשינויים. אין להסיק מכך שאלה הם השלבים לביצוע הפעולות ו/או שזהו הסדר בו השינויים צריכים להתבצע. עליכם לממש את הפעולות בצורה היעילה ביותר כך שתכונות רשימת מערך הקומות תשמרנה.

1. נניח שבפונקציית ה- $main$ התבצעה סדרת הפעולות הבאה:

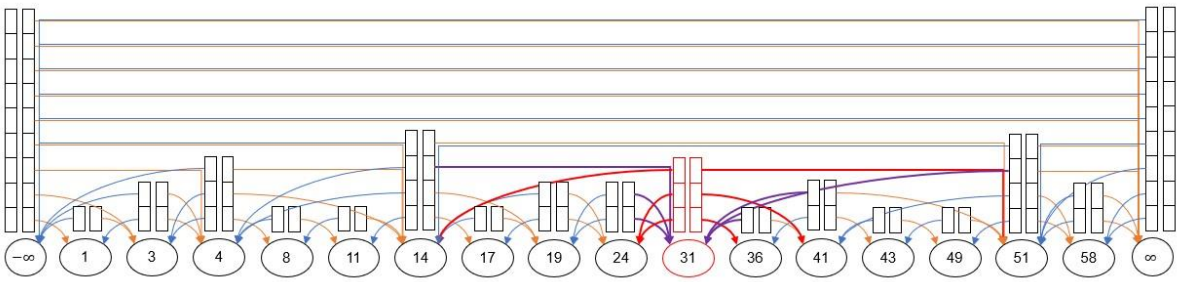
$StoreysArrayList\ list = new\ StoreysArrayList(20);$

$list.insert(1,1);$	$list.insert(24,2);$
$list.insert(3,2);$	$list.insert(31,3);$
$list.insert(4,3);$	$list.insert(36,1);$
$list.insert(8,1);$	$list.insert(41,2);$
$list.insert(11,1);$	$list.insert(43,1);$
$list.insert(14,1);$	$list.insert(49,1);$
$list.insert(17,1);$	$list.insert(51,4);$
$list.insert(19,2);$	$list.insert(58,2);$

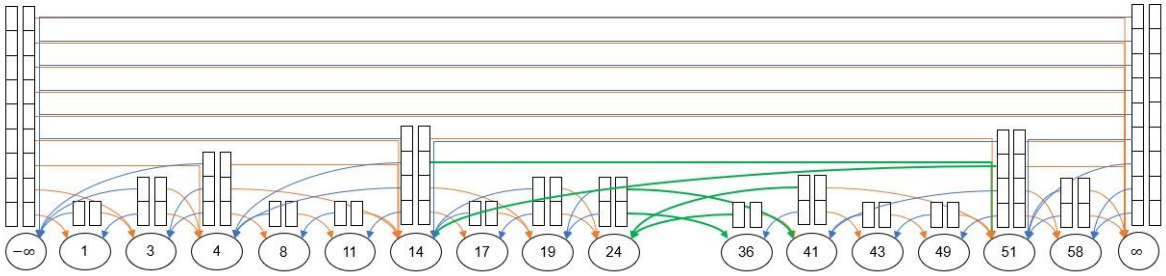
התקבל מבנה עם 16 איברים. שימו לב, גם אם סדר 16 הפעולות האחרונות (כולן חוץ מהאיתחול) היה שונה, בסופן היה מתקבל אותו מבנה.
בכחול: מצביעים אחורה.
בכתום: מצביעים קדימה.
שימו לב, באיורים הבאים מערכי המצביעים בחוליות של המפתחות אינסוף ומינוס אינסוף קטנים יותר ממה שהם צריכים להיות לפי ההגדרה של המבנה (אך לא חסר באיור מידע הכרחי).



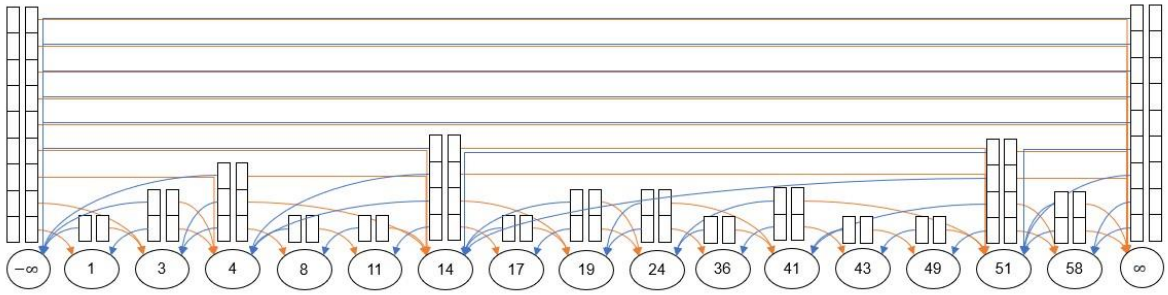
2. נסיר את החוליה עם מפתח "31": $list.remove(list.lookup(31))$
באדום: החוליה (כולל מערכי המצביעים והמצביעים) שיוסרו בעקבות הפעולה
בסגול: המצביעים שישתנו (למשל, ברגע שנסיר את החוליה של 31, החוליה של 41 לא תוכל עוד להצביע אליה).



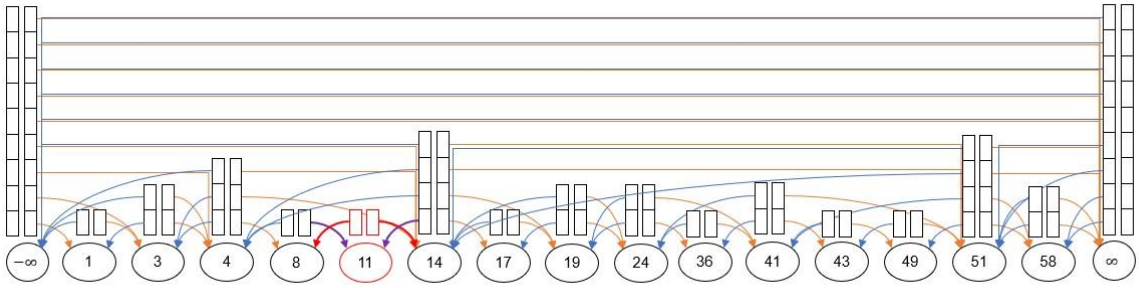
3. המצב לאחר ההסרה.
בירוק: המצביעים שהשתנו.



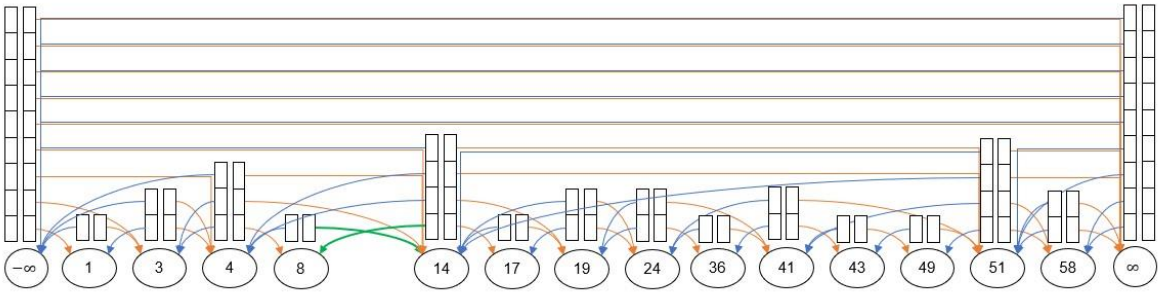
4. כך נראת הרשימה בסוף הפעולה:



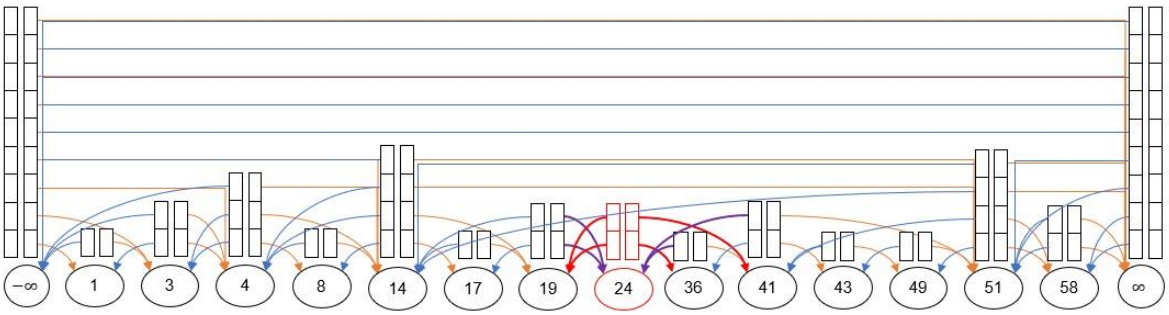
5. כעת נסיר את "11": $list.remove(list.lookup(11))$
באדום: החוליה (כולל מערכי המצביעים והמצביעים) שיוסרו בעקבות הפעולה
בסגול: המצביעים שישתנו.



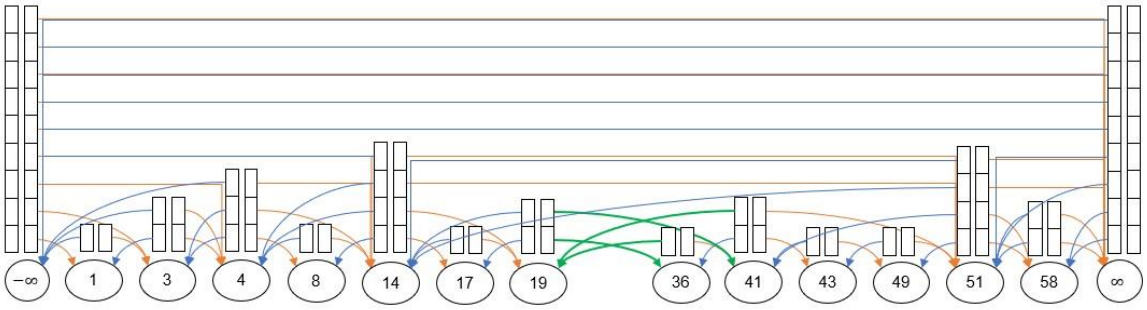
6. המצב לאחר ההסרה.
בירוק: המצביעים שהשתנו.



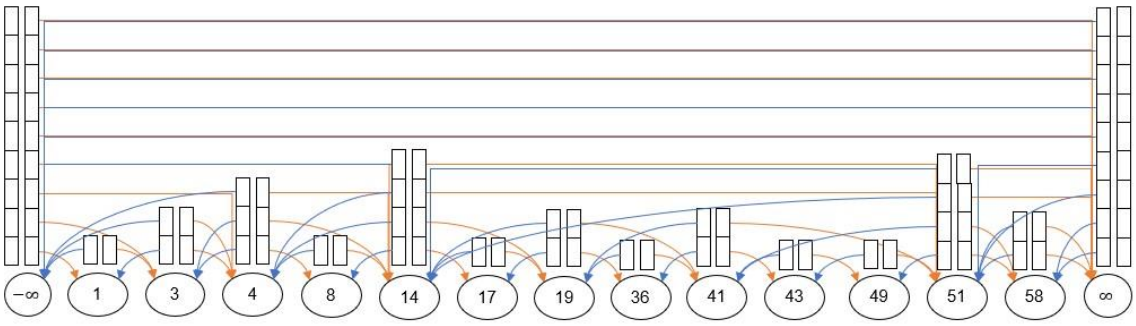
7. כעת נסיר את "24": $list.remove(list.lookup(24))$
באדום: החוליה (כולל מערכי המצביעים והמצביעים) שיוסרו בעקבות הפעולה
בסגול: המצביעים שישתנו.



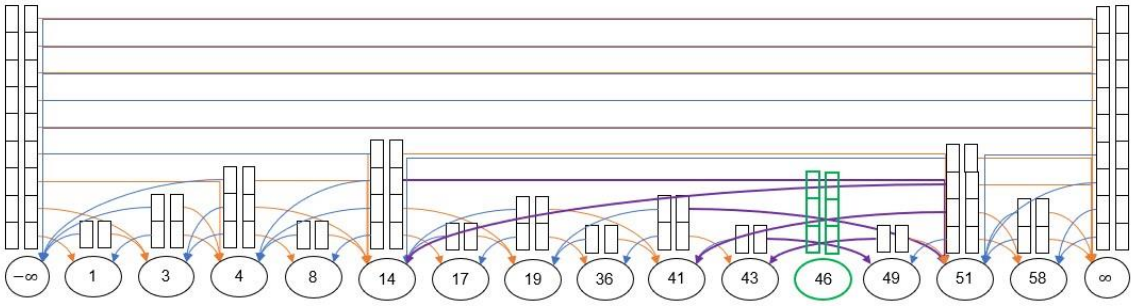
8. המצב לאחר ההסרה.
בירוק: המצביעים שהשתנו.



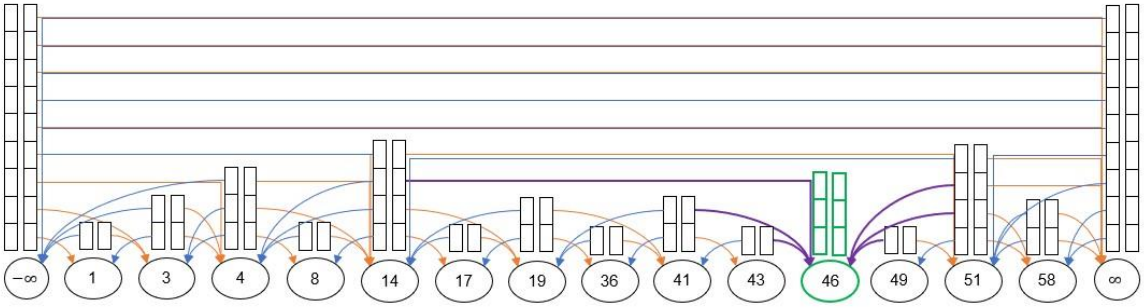
9. כך נראת הרשימה בסוף הפעולה:



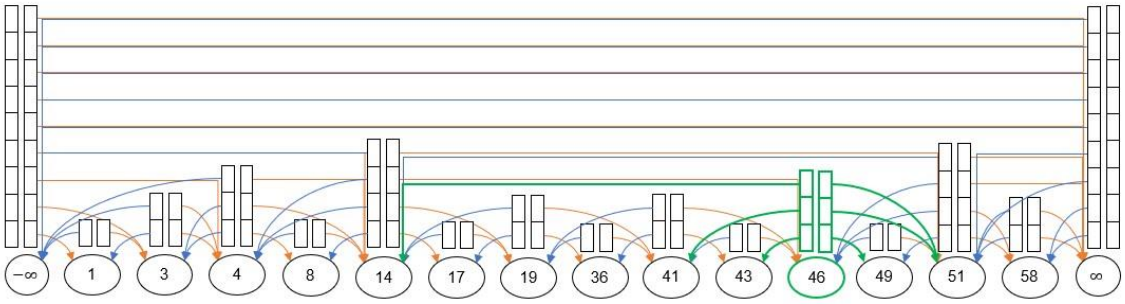
10. כעת נוסיף את המפתח 46 עם מערכי מצביעים בגודל 3: $list.insert(46,3)$
בירוק: החוליה ומערכי המצביעים (בלי המצביעים) שנרצה להוסיף, במקום שבו נרצה להוסיף אותם.
בסגול: המצביעים שישתנו (למשל החוליה עם מפתח 49 לא תצביע עוד לחוליה עם מפתח 43).



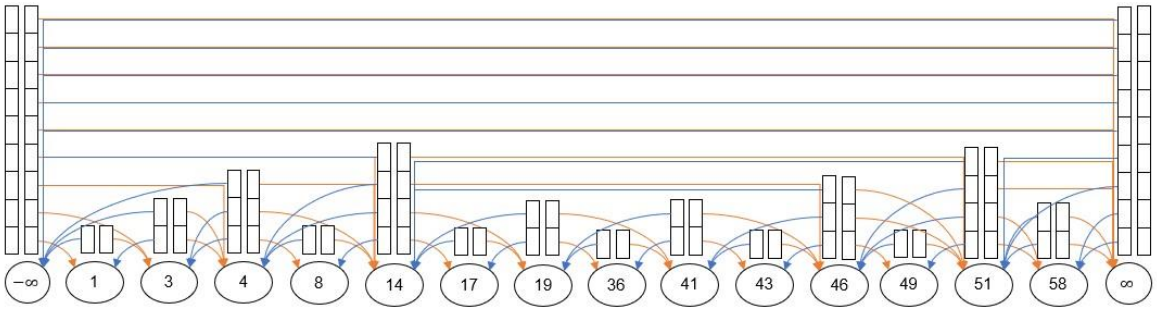
11. שינוי של המצביעים הקיימים.
בסגול: המצביעים שהשתנו (למשל, החוליה עם מפתח 49 מצביעה כרגע לחוליה החדשה).



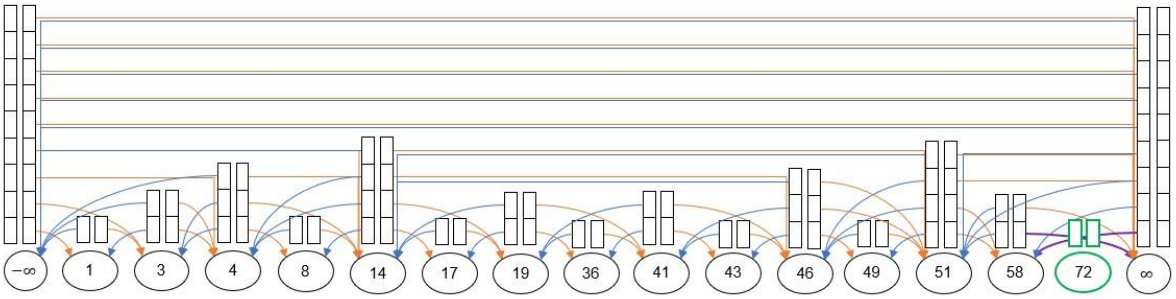
12. הוספת מצביעים מהחוליה החדשה.
בירוק: המצביעים החדשים.



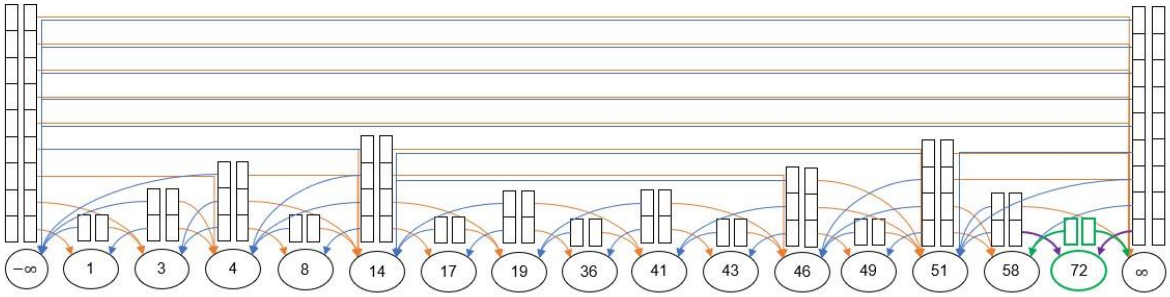
13. כך נראת הרשימה בסוף התהליך:



14. כעת נוסיף את המפתח 72 עם מערכי מצביעים בגודל 1: $list.insert(72,1)$
בירוק: החוליה ומערכי המצביעים (בלי המצביעים) שנרצה להוסיף, במקום שבו נרצה להוסיף אותם.
בסגול: המצביעים שישתנו.



15. שינוי של המצביעים הקיימים, והוספת מצביעים מהחוליה החדשה.
בסגול: המצביעים שהשתנו (למשל, החוליה עם מפתח 49 מצביעה כרגע לחוליה החדשה).
בירוק: המצביעים החדשים.



16. כך נראת הרשימה בסוף התהליך: