

שאלה 1

ד"ר סדר מהפקולטה למתמטיקה התגאה מהעבודה המוצלחת שלכם עם רצפים בתרגיל הבית הקודם והחליט להמשיך שיתוף פעולה עם הקורס שלנו.

הוא חקר את שפת תכנות פרולוג ומאוד התלהב ממנה, על כן ד"ר סדר רוצה שתעזרו לו לבנות מיון של מספרים שלמים בה כחלק של המחקר שלו ב-"**מבנים בינומיאלי**".

נתחיל מכמה הגדרות שסופקו ע"י ד"ר סדר:

- עץ מקיים את תכונת **הערימה** אם המספר שרשום בכל צומת פנימית קטן יותר מכל המספרים שמתחתיו. לשם פשטות נניח שכל הערכים שבעץ הם **שונים**.

- **עץ בינומיאלי** הוא עץ שמקיים את תכונת הערימה:

- עץ בינומיאלי מסדר 0 מכיל צומת אחת בלבד.

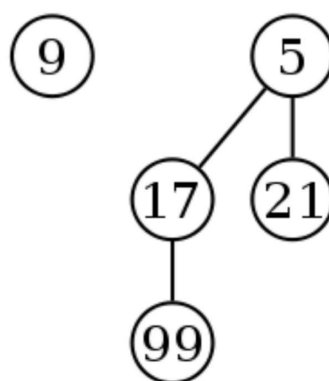
- עץ בינומיאלי מסדר k מכיל בסה"כ 2^k צמתים. צומת אחד בשורש, ועוד $2^k - 1$

צמתים המסודרים באופן הבא: **הבן הראשון הוא עץ בינומיאלי מסדר $k-1$, הבן**

השני הוא עץ בינומיאלי מסדר $k-2$, ואילו הבן ה- k הוא עץ בינומיאלי מסדר

0.

דוגמה לעץ מסדר 0 אשר מכיל מספר 9 ולעץ מסדר 2:



סעיף א'

ממשו פרדיקט $bt/2$ אשר מייצג **עץ בינומיאלי** ומורכב משני פרמטרים:

1. ערך הצומת בשורש.

2. רשימה (שיכולה להיות ריקה) של העצים הבינומיאליים שהם בנוי, כלומר כל פריט

ברשימה הוא פרדיקט bt בעצמו.

ביתן להניח כי שני הפרמטרים קשורים.

הקפידו לבדוק כי כל ההגדרות של עצים בינומיאליים מתקיימים עבור bt .

דוגמה:

```
?- bt(5, [bt(7, [])]).
```

```
true.
```

```
?- bt(7, [bt(5, [])]).
```

```
false.
```

סעיף ב'

לפי ד"ר סדר, בהינתן שני עצים בינומיאליים, מסדר k , קל ליצור מהם עץ בינומיאלי מסדר $k+1$. הוספת עץ אחד כבן אחרון לעץ האחר, תעשה זאת. כדי לשמור על תכונת הערימה, נבחר תמיד להוסיף את העץ שהערך בשורש שלו גדול יותר, כבן נוסף לצומת שהערך בשורש שלו קטן יותר.

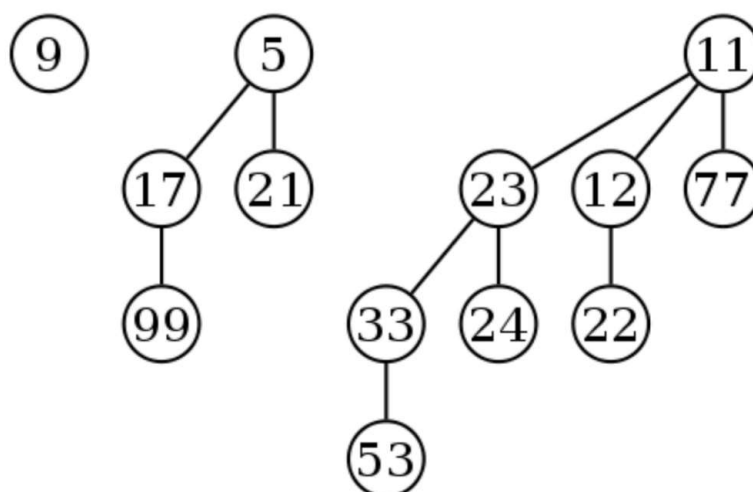
ממשו פרדיקט `merge_bt/3` אשר נכון אמ"מ שני הפרמטרים הראשוניים הם עצים בינומיאליים מאותו סדר והפרמטר השלישי הוא תוצאת המיזוג של שני העצים לפי השיטה הנ"ל.

ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים.
דוגמה:

```
?- merge_bt(bt(3, [bt(4, [])]), bt(5, [bt(7, [])]), X).  
X = bt(3, [bt(5, [bt(7, [])]), bt(4, [])]).
```

סעיף ג'

נגדיר מבנה בינומיאלי חדש. ערימה בינומיאלית היא רשימה של פריטים, כך שבמקום ה- i יכול להופיע עץ בינומיאלי מסדר i או פריט ריק. בפרט, מבנה ערימה בינומיאלית שבה n מספרים בסה"כ נקבע בדיוק על פי הייצוג בביטים של n . הנה לדוגמה בערימה שיש בה 13 מספרים תהיה רשימה שבה העצים הבינומיאליים הבאים:



עצים אלו יופיעו במקומות 0, 2 ו-3 ברשימה. המקום ה-1 ברשימה הוא ריק. שימו לב לכך שהייצוג של 13 בבינארי הוא 0b1101. עוד שימו לב לכך שהמבנה הטופולוגי של כל הערימות בגודל 13 הוא זהה. רק התוכן שלהן שונה. חיבור של שתי ערימות בינומיאליות נעשה בדרך שבה מחברים מספרים שלמים הכתובים בייצוג בינארי.

בפרולוג, נייצג **ערימה בינומיאלית** כרשימה של `bt/2` או אטומים `empty/0` **כאשר האיבר האחרון ברשימה הוא לא `empty`**.

ממשו פרדיקט `add_bt/3` אשר נכון אמ"מ הפרמטר הראשון הוא עץ בינומיאלי ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר הפרמטר השלישי הוא כמו הערימה בפרמטר השני, אך עם הוספת העץ הבינומיאלי.

ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- add_bt(bt(5, []), [bt(4, [])], X).  
X = [empty, bt(4, [bt(5, [])])].
```

סעיף ד'

ממשו פרדיקט `add/3` אשר נכון אמ"מ הפרמטר הראשון הוא מספר ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר הפרמטר השלישי הוא כמו הערימה בפרמטר השני, אך עם הוספת המספר.

ניתן להניח כי שני הפרמטרים הראשוניים הם קשורים. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- add(3, [bt(4, [])], X).  
X = [empty, bt(3, [bt(4, [])])].
```

סעיף ה'

ממשו פרדיקט `pop_min/3` אשר נכון אמ"מ הפרמטר הראשון הוא מספר ושני הפרמטרים האחרים הם ערימות בינומיאליות כאשר המספר הוא האיבר הקטן ביותר בערימה הראשונה והערימה השנייה היא כמו הערימה הראשונה שהוציאו ממנה את האיבר הקטן ביותר.

ניתן להניח כי הפרמטר השני קשור. אין צורך לבדוק כי הפרמטר השני ערימה בינומיאלית.

דוגמה:

```
?- fetch_min(X, [empty, bt(3, [bt(4, [])])], Y).  
X = 3,  
Y = [bt(4, [])].
```

סעיף ו'

הגענו לחלק המרתק לפי מילי ד"ר סדר! בסעיף זה אתם מתבקשים לממש מיון של מספרים שלמים בעזרת ערימות בינומיליות. פתרון אשר משתמש בשיטות מיון אחרות לא יתקבל.

ממשו פרדיקט `sort_me/2` אשר נכון אמ"מ שני הפרמטרים הם רשימות של מספרים שלמים כאשר הרשימה השנייה היא הפרמוטציה הממוינת של הרשימה הראשונה. ניתן להניח כי הפרמטר הראשון קשור וכי אין מספרים חוזרים ברשימות.

דוגמה:

```
?- sort_me([5, -2, 10, 14, 1, 4, 0], X).  
X = [-2, 0, 1, 4, 5, 10, 14].
```