**מגישים:** 204909139, dorfuchs, דור פוקס ו- 212153266, alirayan, עלי ריאן

**קובץ תיעוד – פרויקט מעשי מספר 2 מבני נתונים:**

**מחלקת HeapNode:**

**O(1):**

1. בנאי: מבצע אתחול ל- key.
2. Setters and getters: getters מחזירים מצביע/ערך של השדה המבוקש. setters מקבלים ערך ומאתחלים את השדה המבוקש לערך זה. בנוסף, ה- setters גם מאתחלים את השדה הסימטרי ( parent -> child או next -> prev) של הצומת השני בהנחה וה- setter הוא לשדה שערכו צומת.
3. link: מתודה המקבלת שני צמתים, בודקת למי המפתח הקטן יותר ואז הופכת את הצומת הגדול יותר לבן של הצומת הקטן (בפרט הבן הראשון שאליו מוגד המצביע).
4. cutNode: מתודה שחותכת צומת מההורה שלו ומכניסה אותו כעץ חדש לערמה. בנוסף, היא מעדכנת את הסימון של הצומת ושל אביו.

**O(log(n)):**

1. deleteNode: מתודה המוחקת צומת מהערמה באופן מלא. תחילה היא בודקת אם יש לו ילדים. אם יש, היא עוברת בין הילדים, תוך אתחול שדה ההורה שלהם ל- null וספירתם על מנת לחשב כמה עצים חדשים יתווספו לערמה (O(log(n)) מכיוון שהוכחנו בכיתה שזהו החסם העליון על כמות הילדים של כל צומת). לבסוף, המתודה מכניסה את הרשימה המקושרת של הילדים למיקום בוא ישב הצומת שנמחק ומוחקת את כל מצביעו.

**מחלקת FibonacciHeap:**

**O(1):**

1. isEmpty: מתודה המחזירה true אם הערימה ריקה (בודקת אם גודלה הוא 0) ו- false אחרת.
2. insert: מתודה המקבלת ערך שלם, יוצרת צומת בעל הערך הזה ומכניסה אותו בתור הצומת הראשון בערמה. אם הערימה ריקה היא מכניסה אותו, מאתחלת אותו למינימום וכן את שאר שדות הערימה כגון: size, numOfTrees, ו- marked.