**תרגיל בית מעשי 2 – תיעוד פונקציות**

**עומר פלביץ 206840126  
אלעד שבא 207909409**

מחלקת FibonacciHeap

getFirst – מחזיר את האיבר הראשון בערימה O(1)

getLast – מחזיר את האיבר האחרון בערימה O(1)

isEmpty – בודק האם הערימה ריקה לפי איבר המינימום שחייב להיות בכל ערימה בעלת איבר אחד. O(1)

deleteMin – מוחקת את איבר המינימום. תחילה נבדוק האם איבר המינימום הוא האיבר היחיד – אם כן נהפוך את הערימה לריקה. אחרת, נבדוק אם יש לו בן. אם יש לו, נחבר את הבן כשורש חדש ונחפש מינימום חדש. אם אין לו בן, נמחק אותו ונחבר בין האח הימני שלו לאח השמאלי שלו. בסוף נעשה consolidate . סה"כ זמן הריצה במקרה הגרוע הוא O(n), ואמורטייז O(log n ), כפי שראינו בכיתה.

Meld – מחברת בין שתי ערימות, כך שהערימה השנייה מתחברת לערימה הראשונה. הפונקציה בודקת אם הערימה ה2 ריקה, אם כן משאירה את הערימה הראשונה איך שהיא. אם היא לא ריקה, נחבר את האיבר הראשון בערימה הראשונה לאיבר האחרון בשנייה, נתאים את הגדלים וזהו. סך הכל ביצענו פעולה של חיבור צמתים, זמן הריצה הוא O(1).

Max\_rank – פונקציית עזר, מחזירה את הrank הגבוה ביותר בערימה. עוברת על השורשים ובודקת rank. O(n).

countersRep – מחזירה מערך בו כל כניסה במערך הוא כמות העצים בדרגה המתאימה לכניסה. הפונקציה בודקת האם הערימה ריקה ואם כן מחזירה מערך ריק. אם לא, עוברת על השורשים ומוסיפה 1 לכל מקום במערך בו יש שורש עם הדרגה המתאימה. סה"כ עוברים במקרה הגרוע O(n) שורשים (גם בפוקנציית עזר max\_rank) לכן זהו הזמן ריצה.

decreaseKey – מורידה מצומת מסוים מספר שלם מהמפתח. לאחר מכן מארגנת את העץ בהתאם. הפונקציה בודקת האם הצומת הוא שורש. אם הוא שורש אז הוא יישאר במקומו, נבדוק אם הוא נהיה המינימום ואם כן נחליף את המינימום בהתאם. אם הוא לא שורש, נבדוק האם הוא עדיין גדול מההורה שלו. אם כן – לא נעשה כלום. אם הוא יותר קטן, נבצע חיתוך בעזרת פונקציות העזר. במקרה הגרוע החיתוכים יכולים לעלות ולהגיע עד O(n), אבל בזמן אמורטייז זה O(1).

Potential – פונקציה המחזירה פוטנציאל. מאחר ואנחנו מחזיקים ומתחזקים שדות של מספר העצים ומספר הצמתים המסומנים זה יהיה רק קריאה של שדות ולכן O(1).

totalCuts – מחזיר את מספר החיתוכים. זהו שדה שמתחזקים לכן O(1).

Cuts+cascadingCut – פונקציות שחותכות צמתים כפי שנראה בשיעור. כפי שראינו זמן הריצה הוא במקרה הגרוע O(n) וO(1) אמורטייז.

Consolidate + connect\_same פונקציות עזר לdelete min שמטרתן לאחד את העצים הזהים לעצים גדולים יותר. ניצור מערך של rank של עצים שבכל מקום במערך נאחסן עץ אם הוא בrank של המקום במערך. אם יהיו שני עצים באותו מקום נשתמש בconnect ונאחד בינהם.

מחלקת HeapNode

נחזיק שדות של מפתח, בן, הורה, אח ימני ושמאלי, האם הצומת מסומן ומה הrank שלו. בנוסף שדה של מיקום.

לכן כל הget של כל השדות האלה הוא O(1).