**רטוב 1 – חלק יבש:**

מבני נתונים:

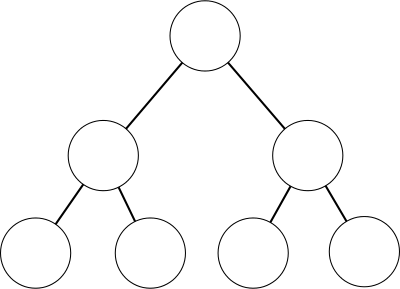
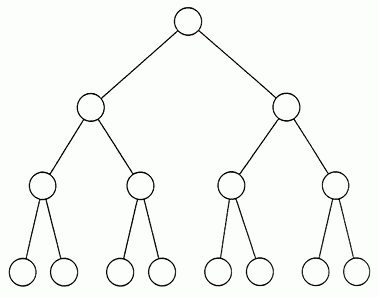
* - כפי שלמדנו בהרצאה ובתרגול, מימשנו עץ AVL גנרי אשר מחזיק מפתח ומידע ( , ), ומאפשר פעולות כגון חיפוש, הסרה והוספה בסיבוכיות הנדרשת בתרגיל. העץ דורש גם מחלקת Comparer אשר לה אופרטור () אשר ישווה בין טיפוסי .
* *- טיפוס אשר מגדיר עובד , מכיל בתוכו נתונים כגון מזהה עובד ודירוגו ומצביע לחברה שמעסיקה אותו.*
* - טיפוס אשר מגדיר חברה , מכיל בתוכו נתונים כגון מזהה החברה וגם מחזיק את כל העובדים של החברה בצורה המתוארת בהמשך.
* - מכיוון שדירוג העובד אינו מזהה חד חד ערכי של עובד , יצרנו את אשר מכיל בתוכו את דירוג העובד ואת המזהה הייחודי שלו . מחלקה זו תאפשר לנו לבנות עץ שמכיל בתוכו פוסטים וממוין לפי דירוג בצורה ראשית ולפי מזהה עובד בצורה משנית.

המערכת :

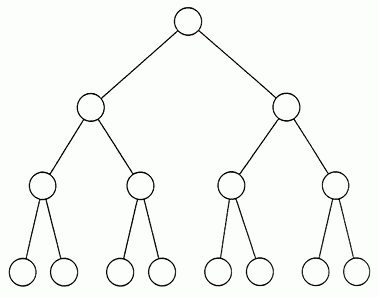
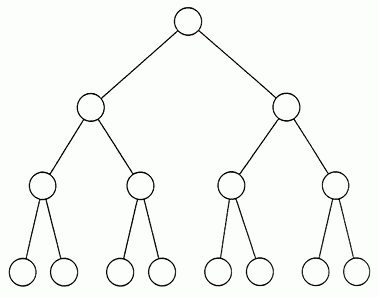
* עץ עובדים ממוין לפי .
* עץ עובדים ממוין לפי .
* עץ חברות ממוין לפי .
  + עץ עובדי החברה ממוין לפי .

מבנה זה מאפשר לנו לבצע חיפושים בסיבוכיות בהינתן וגם להחזיר את העובדים ממוינים לפי הדירוג שלהם .

בנוסף לכל זה , נשמור בעבור כל חברה את שלה , העובד בעל הדירוג הגבוה ביותר בחברה וגם כללי של כל המערכת , העובד בעל הדירוג הגבוה ביותר בכל המערכת .

שרטוט מבני הנתונים:

ZOOM IN



פירוט הפעולות:

- פעולה זו מאתחלת את מבנה הנתונים אשר מאתחל 3 עצי ראשיים (( , מספר קבוע של פקודות אשר אינו תלוי באף משתנה ולכן סיבוכיות הריצה הינה .

- פקודה זו מוסיפה עובד למערכת, לאחר בדיקת חוקיות מזהה העובד נבצע בדיקה אם העובד קיים כבר במערכת (חיפוש בעץ - ). במידה והעובד לא קיים ניצור עובד חדש ונכניס אותו ל2 עצי העובדים של המערכת ( כל הכנסה ) . סה"כ נבצע חיפוש והכנסה ולכן .

- פקודה זו מוסיפה חברה למערכת, לאחר בדיקת חוקיות מזהה החברה נבצע בדיקה אם החברה קיימת כבר במערכת (חיפוש בעץ - ). במידה והחברה לא קיימת ניצור חברה חדשה ונכניס אותה לעץ החברות של המערכת ( כל הכנסה ) . סה"כ נבצע חיפוש והכנסה ולכן .

-

פקודה זו מוסיפה עובד עם מזהה לחברה . לאחר בדיקת חוקיות הקלטים נחפש את העובד במערכת ואת החברה במערכת ( ). נגדיר לעובד את החברה כחברה שאליה הוא שייך . נוסיף את העובד לעץ העובדים של החברה ( .*סה"כ:* ( )*.*

- פעולה זו מוחקת עובד מהמערכת, כאשר לכל עובד יש מקסימום 3 עצים שהוא צריך להימחק מהם. נבדוק שקיים עובד כזה במערכת , במידה וכן נביא את החברה שלו ( ), ובתוכה נמחק מהעץ עובדים את העובד המתאים , בנוסף נמחק מ2 עצי העובדים הראשיים את העובד , כל מחיקה תעלה ,בסה"כ הפעולה תהייה בסיבוכיות של .

פקודה זו ממזגת בין 2 חברות, בכך שיוצרת חברה חדשה בעלת מזהה החברה הגדולה יותר(מבחינת מספר עובדים),מעבירה את העובדים מ2 החברות אל החברה החדשה (רק את העובדים שדירוגם גבוה מדירוג הסף המתקבל כפרמטר) ומסירה את החברה הקטנה יותר.

נמצא את כל אחת מהחברות בעץ החברות במערכת-בסיבוכיות של -חיפוש בעץ ,כאשר הוא מספר החברות בעץ החברות. לאחר מכן נחליט מי החברה הגדולה יותר כדי לקבוע את מספר הזיהוי של החברה הממוזגת(סיבוכיות (כעת נשלוף את העובדים מעצי העובדים בכל אחת מהחברות הנ"ל, ע"י סיור in order ,ונשמור אותם במערכי עזר ממוינים.(בגלל הסיור in-order)-סיבוכיות כפי שנלמד בתרגול עבור סיורי in-order.

נעבור על מערכי העובדים על מנת לספור את מספר העובדים שדירוגם לא עולה על סף הדירוג המינימלי.

ניצור מערכי עובדים חדשים לכל חברה ואליה נעתיק רק את העובדים שדירוגם עולה על הסף המינימלי- סה"כ סיבוכיות של עבור כל המעברים על מערכי העזר.

ניצור מערך עובדים בגודל סה"כ העובדים שמיועדים להיות בחברה הממוזגת, ונבצע מיזוג בין 2 מערכי העובדים אל המערך הסופי בסיבוכיות של כפי שנלמד בתרגול.

ניצור עץ שלם/כמעט שלם עם מספר העובדים בחברה הממוזגת, עפ"י האלגוריתם שנלמד בהרצאה -

נמחק את העלים המיותרים בעץ החדש, כך שנשאר עם מספר צמתים כולל כגודל מספר העובדים, נבצע סיור in-order ונמקם את איברי מערך העובדים החדש בצמתי העץ-סיבוכיות של

סה"כ סיבוכיות :

- *פקודה זו משנה את של העובד . תחילה נמצא אם קיים כזה עובד ( ) אם העובד נמצא נקרא לפונקציה שמוחקת אותו מהמערכת ואז נוסיף אותו מחדש באמצעות קריאה לפונקציה עם הדירוג החדש , סה"כ הפעולות האלה ייקחו כל אחת . ולכן סה"כ הפעולות יהיו בסיבוכיות של .*

- מאחר ובהוספת, מחיקת ועדכון עובדים מתבצע עדכון של המשתנה , מובטח שלכל חברה המשתנה הנ"ל יהיה עדכני ויכיל את העובד בעל הדירוג הגדול ביותר כנ"ל גם למשתנה הכללי של המערכת. ולכן בO(1) אנו שולפים את המשתנה הנדרש ומחזירים אותו.

- פקודה זו מחזירה עובדים לפי דירוגם לתוך מערך ומחזירה את כמות העובדים האלה.

במידה וקיבלנו מזהה חוקי הפקודה תחזיר את העובדים ממויינים לפי דירוג של אותה חברה , במידה והערך שלילי יוחזרו כל העובדים במערכת.

אפשרות אחת – כל הטרולים במערכת :

קודם ניצור את המערך לפי כמות הנודים שיש במערכת . לאחר מכן נרוץ על כל העובדים במערכת בעץ שכבר ממויין לפי דירוג . כל עובד נוסיף למערך שיצרנו, מעבר כזה על כל הפוסטים הוא . לסיום נחזיר את המערך המעודכן ואת מספר העובדים. לסיכום סיבוכיות הזמן במעבר על כל המערכת הוא

אפשרות שנייה – חברה תקינה :

במקרה הנ"ל קודם נחפש את החברה המבוקשת , חיפוש כזה הוא . לאחר מכן ניצור את המערך לפי כמות העובדים שיש לאותה חברה . נרוץ על כל העובדים של אותה החברה שמצאנו בעץ שכבר ממוין לפי דירוג העובדים . כל עובד נוסיף למערך שיצרנו , מעבר כזה על כל העובדים הוא . לסיום נחזיר את המערך המעודכן ואת מספר העובדים לסיכום סיבוכיות הזמן במידה וטרול קיים ותקין היא

- פעולה זו משחררת את הזיכרון של כל הנתונים שנשמרו עד כה במערכת. יש למחוק את עץ החברות ועץ העובדים של כל חברה, את 2 עצי העובדים

מחיקת כל חברה תבצע מחיקת עץ העובדים שלה– זה יתבצע ב לכל היותר.

מחיקת עץ החברות תתבצע ב-.

מחיקת 2 עצי העובדים תתבצע ב-.

ולכן לסיכום פעולת Quit תתבצע בסיבוכיות .