הבנה שלנו יורכב מ 3 מבנים עיקריים :

1. מבנה של Employee, והוא מכיל השדות הבאים:

Int id;

Int company\_id;

Int grade;

Int salary;

1. מבנה של Company, והוא מכיל השדות הבאים:

Int id;

Double value;

Hashtable employees\_with\_zero\_salary;

Int grade\_bump\_for\_zero\_salary;

Int number\_of\_employees\_with\_no\_salary;

Int sum\_of\_grades\_with\_no\_salary;

Ranktree employees\_with\_salary;

1. מבנה של Industry, והוא מכיל השדות הבאים:

Int number\_of\_companies;

UnionFind companies;

// Hashtable for company owners!

Hashtable employees\_with\_zero\_salary;

Int grade\_bump\_for\_zero\_salary;

Int number\_of\_employees\_with\_no\_salary;

Int sum\_of\_grades\_with\_no\_salary;

Ranktree employees\_with\_salary;

העץ שלנו מחזיק: דרגה,סכומים(כמה בנים בתת העץ),סכום ה grades בתת העץ, ו bump לבנים בתת בעץ.

Init(int k) –

נבדוק את הערך של k ואם הוא קטן/שווה לאפס אז נחזיר FAILURE. אחרת , ניצור UF עם k חברות ב O(k) כפי שנלמד בכיתה עם עץ לעובדים וטבלת ערבול אשר ניצור אותם ב O(1) ובסוף נחזיר את המבנה של Industry ב O(k) ונסיים כנדרש!

addEmployee((void \*DS,int EmployeeID, int CompanyID, int Grade) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

נחפש את העובד בעל מזהה EmployeeID ב Hashtable של כלל העובדים במבנה ב O(1) ואם נמצא אותו אז נחזיר שגיאת FAILURE Uנסיים ב O(1) בממוצע.

נחפש את החברה בעלת מזהה CompanyID ב O(log\*(k)) משוערך כפי שנלמד בכיתה,ונסמנה company\_to\_find, ואז נכניס את העובד החדש ל Hashtable של העובדים במבנה של החברה עצמה וגם כן למבנה הכללי של Industry O(1) בממוצע.ונבצע ההוספות הבאות :

Company\_to\_find->number\_of\_employees\_with\_no\_salary++

Company\_to\_find->sum\_of\_grades\_with\_no\_salary+=Grade

DS->number\_of\_employees\_with\_no\_salary++

DS->sum\_of\_grades\_with\_no\_salary+=Grade

וכך נסיים ההכנסה ב O(log\*(k)) משוערך בממוצע כנדרש!

removeEmployee(void \*DS, int EmployeeID) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

נחפש את העובד בעל מזהה EmployeeID ב Hashtable של כלל העובדים במבנה ב O(1) ,ונסמנו emp,ואם לא נמצא אותו אז נחזיר שגיאת FAILURE ונסיים ב O(1) בממוצע.

אם כן מצאנו אותו אז נחפש את החברה אליה הוא שייך ב O(1) בממוצע בטבלת הערבול של הקבוצות,ונסמנה company\_to\_find, נוציא אותו מטבלת הערבול של העובדים בחברה וגם במבנה הכללי של Industry ב O(1) בממוצע, ואז אם יש לו salary > 0 נוציא אותו מעץ הדרגות של העובדים בחברה ובמבנה הכללי ב O(log(n)).

ואז נבדוק אם ה salary שלו היא 0 ואם כן, אז נבצע :

Company\_to\_find->number\_of\_employees\_with\_no\_salary--

Company\_to\_find->sum\_of\_grades\_with\_no\_salary-=emp->Grade

DS->number\_of\_employees\_with\_no\_salary--

DS->sum\_of\_grades\_with\_no\_salary-=emp->Grade

acquireCompany(void \*DS, int AcquirerID, int TargetID, double Factor) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.וגם נחפש את החברה של AcquirerID ושל TargetID ב UF של החברות ב O(log\*(k)) משוערך כפי שנלמד בכיתה,ואם נמצא שהן שייכות לאותה קבוצה אז נחזיק INVALID\_INPUT.

נחפש את שתי החברות ב O(log\*(k)) משוערך ב UF,ונסמנם target ו acquirer.

ולאחר מכן נמזג את שתי טבלאות הערבול של העובדים בחברות לתוך טבלה אחת שנשים אותה בתוך החברה של acquirer.ו נמזג גם את שני העצים של העובדים לתוך עץ אחד כפי שנלמד בכיתה וכל זה מתבצע ב בממוצע(כמובן על הדרך משנים את החברה של העובדים להיות acquirer). ולאחר מכן נסיר את החברה של target מתוך הטבלת ערבול של החברות ב O(1) בממוצע.

וכך נסיים הפעולה ב כנדרש.

employeeSalaryIncrease(void \*DS, int EmployeeID, int SalaryIncrease) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

נחפש את העובד בעל מזהה EmployeeID ב Hashtable של כלל העובדים במבנה ב O(1) ,ונסמנו emp,ואם לא נמצא אותו אז נחזיר שגיאת FAILURE ונסיים ב O(1) בממוצע.

אם כן מצאנו אותו אז נחפש את החברה אליה הוא שייך ב O(1) בממוצע בטבלת הערבול של הקבוצות,ונסמנה company\_to\_find, ואז אם יש לו salary > 0 נוציא אותו מעץ הדרגות של העובדים בחברה ובמבנה הכללי ב O(log(n)) ונכניס אותו שוב אחרי שנגדיל את ה salary שלו בערך של salaryincrease לשני העצים שהוצאנו אותו מהם ב O(log(n)).

ואם יש לו salary==0 אז נבצע :

Company\_to\_find->number\_of\_employees\_with\_no\_salary--

Company\_to\_find->sum\_of\_grades\_with\_no\_salary-=emp->Grade

DS->number\_of\_employees\_with\_no\_salary--

DS->sum\_of\_grades\_with\_no\_salary-=emp->Grade

ונכניס אותו לעצי הדרגות של החברה ושל המבנה הכללי ב O(log(n)).

וכך נסיים בפעולה ב O(log(n)) בממוצע כנדרש!

promoteEmployee(void \*DS, int EmployeeID, int BumpGrade) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

נחפש את העובד בעל מזהה EmployeeID ב Hashtable של כלל העובדים במבנה ב O(1) ,ונסמנו emp,ואם לא נמצא אותו אז נחזיר שגיאת FAILURE ונסיים ב O(1) בממוצע.

אם כן מצאנו אותו אז נחפש את החברה אליה הוא שייך ב O(1) בממוצע בטבלת הערבול של הקבוצות,ונסמנה company\_to\_find, ואז ואם יש לו salary==0 אז נבצע :

Company\_to\_find->sum\_of\_grades\_with\_no\_salary+=diff

DS->sum\_of\_grades\_with\_no\_salary+=diff

כאשר diff הוא ההפרש בין ה grade החדש לישן.

ואם יש לו salary > 0 נוציא אותו מעץ הדרגות של העובדים בחברה ובמבנה הכללי ב O(log(n)) ונכניס אותו שוב אחרי שנגדיל את ה grade שלו לשני העצים שהוצאנו אותו מהם ב O(log(n)).

sumOfBumpGradeBetweenTopWorkersByGroup (int CompanyID, int m, void \* sumBumpGrade) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

אם companyID > 0 אז נמצא את החברה בעלת מזהה זה ב O(log\*(k)) משוערך כפי שנלמד בכיתה ונסמנה ב company\_to\_find ונפעול על העץ של העובדים בחברה באופן המתואר מטה, אחרת נפעול באותו אופן רק עם העץ של כלל העובדים במבנה.

נחפש את העובד בעץ בעל דרגה m ב O(log(n)) ואם הוא לא קיים אז זה סימן שאין מספיק עובדים במבנה שעונים על התנאים ולפיכך נחזיר FAILURE.

אם העובד הזה קיים אז סימן שיש מספיק ועובדים שעונים על התנאים ולכן נחפש העובד הראשון בתת העץ הימני שיש לו פחות מ m עובדים בתת העץ שלו ונחזיר את האב שלו, שנסמנו node, והחיפוש הזה לוקח O(log(n)) כפי שנלמד בכיתה. ואז החל מ node נבדוק כמה איברים יש בתת העץ שלו ונסמנם sons וגם כן נסמן את סכום הדרגות בתת העץ שלו ב sum ונתחיל לרדת בתת העץ השמאלי עד שנגיע לצומת שיש לו מספיק בנים כך שאם נחסיר אותם מ sons נקבל מספר קטן/שווה ל m. נחסיר את הערכים המתאימים של הצומת שהגענו אליו מ sons ו sum.ואז נוסיף את הערכים של תת הצומת הימני שלו ל sons ו sum עד שנגיע ל sons == m (בכל פעם נוסיף את הערגים של תת העץ הימני ונחסיר את של תת העץ השמאלי באופן דומה למה שעשינו למעלה) ונסיים הפעולה כך ב O(log\*(k) + log(n)) משוערך

averageBumpGradeBetweenSalaryByGroup () –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

אם companyID > 0 אז נמצא את החברה בעלת מזהה זה ב O(log\*(k)) משוערך כפי שנלמד בכיתה ונסמנה ב company\_to\_find ונפעול על העץ של העובדים בחברה באופן המתואר מטה, אחרת נפעול באותו אופן רק עם העץ של כלל העובדים במבנה.

נמצא את האב קדמון של ...

companyValue(void \*DS, int CompanyID, void \* standing) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.

Quit(void \*\*DS) –

bumpGradeToEmployees(void \*DS, int lowerSalary, int higherSalary, int BumpGrade) –

נבדוק שהנתונים שקיבלנו תקינים. אחרת, נחזיר שגיאה מתאימה מסוג INVALID\_INPUT.