

רטוב 2 מבני נתונים



מגישים :

מיכאל שליזרמן 315234351

רבקה ליפקינד 208156596

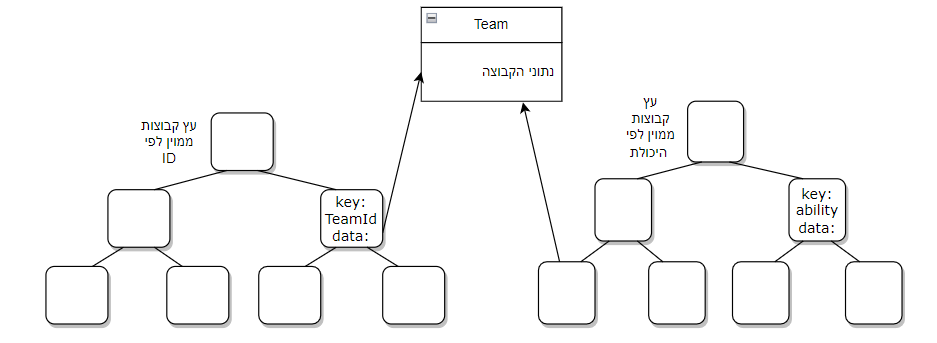
05/01/2022

מבני הנתונים שנשתמש בהם :

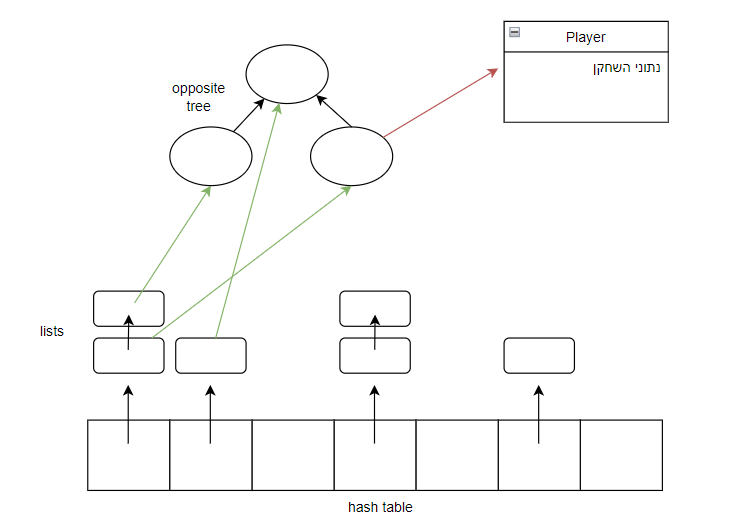
1. **עץ קבוצות ממוין לפי המספר המזהה שלהן -** עץ AVL של קבוצות ממוין לפי teamId. כל איבר בעץ יכיל key שיהיה המספר המזהה של הקבוצה וdata שיחזיק מצביע לקבוצה עצמה.

כל קבוצה מחזיקה : המספר המזהה שלה, מספר הנקודות הכולל שלה, מספר שחקניה הכולל, מספר המשחקים ששיחקה, משתנה בולינאני שאומר האם בקבוצה יש שוער, סכום היכולות של שחקניה ,הרוח הקבוצתית, האם היא עדיין במשחק, מצביע לראש הקבוצה בעץ ההפוך.

1. **עץ דרגות של קבוצות ממוין לפי היכולת של הקבוצה -** כל איבר בעץ יכיל key שייצג את היכולת של הקבוצה וdata שיחזיק מצביע לקבוצה עצמה.



1. **טבלת HASH TABLE –** נמפה בין כל מזהי השחקנים לטבלת ערבול דינמית, תוך שימוש בchain .hashingכלומר , כל תא בטבלת הערבול מצביע לרשימה מקושרת.כל איבר ברשימה המקושרת מחזיק מצביע לאיבר שנמצא בעץ הפוך .
2. **עצים הפוכים -** עץ ההפוך מייצג את קבוצה, וכל איבר בעץ ההפוך מחזיק מצביע לשחקן בקבוצה שהוא מייצג.השורש של בעץ יחזיק בנוסף לזאת גם מצביע לקבוצה אליה שייכים כל השחקנים בעץ זה.  
   בנוסף, כל שורש יחזיק מידע נוסף שישמש אותנו לחישובים מסוימים :  
   א. מכפלת פרמוטציות  
   ב. מס' משחקים
3. **משתנה שיחזיק את מספר הקבוצות במערכת.**



מימוש הפעולות הנדרשות :

פעולה זו מאתחלת את כל העצים והמבנים להיות ריקים. כל אחת מן הפעולות מתבצעת ב O(1)ולכן סיבוכיות זמן הפעולה היא O(1).

עבור כל אחד מן עצי הAVL שאנו משתמשים בהם – באמצעות סיור postorder נבצע מחיקה ושחרור זיכרון של כל איברי העץ. ישנם שני *עצים המכילים קבוצות , והם מכילים לכל היותר איברים ולכן סיבוכיות הסיור הנ''ל היא . (כל קבוצה מופיע לכל היותר ב2 עצים ולכן ) . בנוסף, נבצע מחיקה של איברי הרשימות המקושרות שבhash table והצמתים בעצים ההפוכים שהם מצביעים אליהם. כל איבר ברשימה וכל צומת בעץ מייצג שחקן, לכן אם ישנם שחקנים במערכת, סיבוכיות הפעולה היא .  
סיבוכיות פעולת שחרור הזיכרון היא ולכן סה''כ נקבל שסיבוכיות הפעולה היא .*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *לאחר מכן נחפש קבוצה בעלת המזהה* teamId *בעץ הקבוצות. במידה ומזהה זה קיים בעץ , נחזיר* FAILURE*.   
אחרת, נוסיף את הקבוצה הנ''ל לעץ כאשר נעדכן את מספר שחקניה להיות 0 . נוסיף את הקבוצה גם לעץ שממוין לפי יכולת. במקרה של בעיה בהקצאת הזיכרון ביצירת הקבוצה נחזיר* ALLOCATION\_ERROR. *אם ההוספה צלחה , נחזיר* SUCCESS*.*

*סיבוכיות זמן : כפי שראינו בהרצאות ובתרגולים, הוספה לעץ* AVL *היא ב כאשר הוא מספר הקבוצות במערכת (ומספר האיברים ברגע נתון בעץ הקבוצות).*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *לאחר מכן נחפש קבוצה בעלת המזהה* teamId *בעץ הקבוצות .**במידה ולא קיימת בעץ קבוצה בעלת מזהה זה, נחזיר* FAILURE *. אחרת, הקבוצה מחזיקה מצביע לראש הקבוצה העץ ההפוך. נעדכן אצל ראש הקבוצה את המצביע לקבוצה להיות NULL.  
נמחק את הקבוצה משני עצי הקבוצות ע''י שחרור זיכרון. במקרה של בעיה בשחרור הזיכרון נחזיר* ALLOCATION\_ERROR*. אחרת, נחזיר* SUCCESS.

*סיבוכיות זמן : כפי שראינו בהרצאות ובתרגולים, הסרה מעץ* AVL *היא ב כאשר הוא מספר הקבוצות במערכת (ומספר האיברים ברגע נתון בעץ הקבוצות). שאר הפעולות מתבצעות ב .*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT.. *כעת נחפש את המזהה המייצג את הקבוצה אליה השחקן אמור להצטרף בעץ הקבוצות. אם קבוצה בעלת מזהה זה אינה קיימת בעץ, נחזיר* FAILURE *. בדיקה בhash*

*אחרת, ניצור את השחקן . במקרה של בעיה בהקצאת הזיכרון נחזיר* ALLOCATION\_ERROR *.*

*סיבוכיות זמן :*

* *מציאת הקבוצה אליה שייך השחקן בעץ הקבוצות-*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. נחפש את מזהי שתי הקבוצות בעץ הקבוצות. אם לפחות אחד מהם לא נמצא בעץ, או ששניהם נמצאים אך לפחות לאחד מהם אין שוער, נחזיר FAILURE*. נחשב את יכולתה של כל קבוצה לנצח לפי הנוסחא הנתונה. אם לקבוצה 1 יש יכולת גדולה יותר נחזיר 1, אם לקבוצה 2 נחזיר 3. במקרה של שיוויון , נחשב את הכוח הרוחני באמצעות רוח הקבוצה (משתנה שהקבוצה שומרת). אם הכוח הרוחני של קבוצה 1 גדול יותר נחזיר 2 , ואם של קבוצה 2 נחזיר 4. במקרה של שיוויון נחזיר 0. נעדכן את מספר הנקודות של כל קבוצה בהתאם (ניצחון שווה +3 נקודות לקבוצה המנצחת ותיקו שווה נקודה בודדת לכל אחת מהקבוצות) ונגדיל את מספר המשחקים של כל קבוצה ב1 , ונחזיר* SUCCESS*.*

*סיבוכיות זמן :*

* *מציאת שתי הקבוצות בעץ הקבוצות-*
* *חישוב יכולתה של כל קבוצה וכוחה הרוחני - מכיוון שכל קבוצה מחזיקה את המשתנים הנדרשים לחישוב והחישוב עצמו מתבצע ב*
* *עדכון נתונים -*

*סה''כ סיבוכיות הזמן היא כנדרש.*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *נפעיל את* *פונקציית הHASH על מזהה השחקן ונמצא אותו ברשימה המקושרת . אם הוא לא נמצא ברשימה, נחזיר* FAILURE*.*

*אחרת, נמצא את הצומת של השחקן בעץ ההפוך ונבצע סכימה לאורך המסלול לשורש של שדות הgames. סכימה זו מבטאת את סך המשחקים שקבוצתו המקורית של השחקן שיחקה. ממספר זה נחסיר את מספר המשחקים שקבוצתו המקורית שיחקה טרם הצטרפותו ונוסיף את מספר המשחקים שהוא שיחק בעת הצטרפותו למערכת ונחזיר את התוצאה. בנוסף, נבצע כיווץ מסלול.*

*סיבוכיות זמן :*

* *מציאת מזהה השחקן בטבלת הHASH - בממוצע על הקלט.*
* *מציאת השורש של העץ ההפוך וסכימה לאורך המסלול-* משוערך*.*

*סה''כ :* משוערך *, בממוצע על הקלט.*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *נפעיל את* *פונקציית הHASH על מזהה השחקן ונמצא אותו ברשימה המקושרת . אם הוא לא נמצא ברשימה, נחזיר* FAILURE*. אם הוא נמצא, נוכל לעבור לאיבר בעץ ההפוך שאליו מצביע מקומו ברשימה המקושרת ולהתקדם עד שורש העץ. במידה ושורש העץ מכיל מצביע לקבוצה שהיא NULL , סימן שהשחקן הודח מהטורניר ונחזיר* FAILURE*. אם לא, נעדכן את מספר הכרטיסים של השחקן ונבצע כיווץ מסלול.*

*סיבוכיות זמן :*

* *מציאת מזהה השחקן בטבלת הHASH - בממוצע על הקלט.*
* *מציאת השורש של העץ ההפוך-* משוערך*.*
* *עדכון מספר הכרטיסים של השחקן -*

*סה''כ :* משוערך *, בממוצע על הקלט.*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *נפעיל את* *פונקציית הHASH על מזהה השחקן ונמצא אותו ברשימה המקושרת . אם הוא לא נמצא ברשימה, נחזיר* FAILURE*. אם הוא נמצא, נוכל לעבור לאיבר בעץ ההפוך שאליו מצביע מקומו ברשימה המקושרת , ונחזיר את מספר הכרטיסים של השחקן.*

*סיבוכיות זמן :*

* *מציאת מזהה השחקן בטבלת הHASH - בממוצע על הקלט.*
* *החזרת מספר הכרטיסים של השחקן - (משתנה שהשחקן מחזיק, כל המעברים עד לשחקן הם מעברי מצביעים שמתבצעים ב ).*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT.

*לאחר מכן נחפש קבוצה בעלת המזהה* teamId *בעץ הקבוצות. במידה ומזהה זה אינו קיים בעץ , נחזיר* FAILURE*. אם הוא קיים, נחזיר את מספר הנקודות של הקבוצה.*

*סיבוכיות זמן :*

* *מציאת הקבוצה בעץ הקבוצות-*
* *החזרת מספר הנקודות - (משתנה שהקבוצה מחזיקה).*

*סה''כ :*

*תחילה נבדוק שערכי הפונקציה תקינים (האינדקס המבוקש קטן שווה ממספר הקבוצות במערכת וגדול שווה מ0 ,קיימות קבוצות במערכת) , במידה ולא – נחזיר* FAILURE. *נחפש את האיבר בעל האינדקס i+1 בעץ הדרגות שממוין לפי יכולת כפי שלמדנו בתרגול ונחזיר את מזהה הקבוצה.*

*סיבוכיות זמן :*

* *מציאת האיבר בעץ קבוצות בהינתן האינדקס שלו - בזהה לנלמד בתרגולים.*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. *נפעיל את* *פונקציית הHASH על מזהה השחקן ונמצא אותו ברשימה המקושרת . אם הוא לא נמצא ברשימה, נחזיר* FAILURE*.*

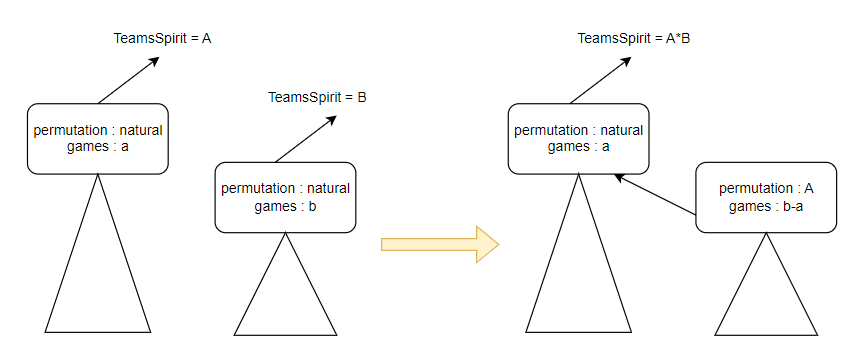
*אחרת, נמצא את הצומת של השחקן בעץ ההפוך ונבצע הרכבה לאורך המסלול לשורש של שדות הpermutation. נבצע כיווץ מסלול. נרכיב את התוצאה בהרכבת של כל הspirit של קודמיו בקבוצתו המקורית (משתנה שהשחקן מחזיק). נחזיר את התוצאה.*

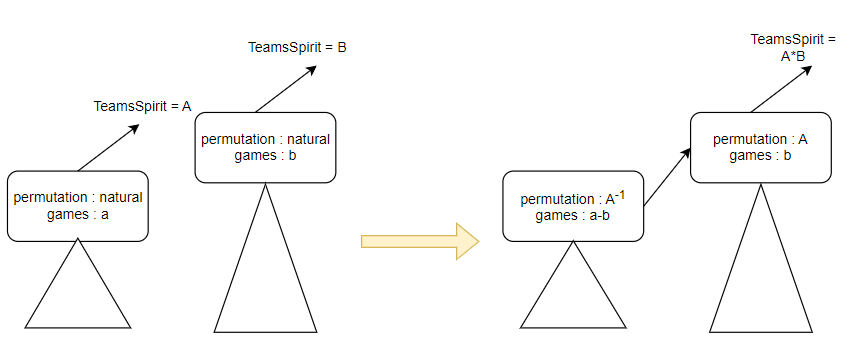
*סיבוכיות זמן :*

* *מציאת מזהה השחקן בטבלת הHASH - בממוצע על הקלט.*
* *מציאת השורש של העץ ההפוך והרכבה לאורך המסלול-* משוערך*.*

*סה''כ :* משוערך *, בממוצע על הקלט.*

*תחילה נבדוק שערכי הפונקציה תקינים , במידה ולא – נחזיר* INVALID\_INPUT. נחפש את מזהי שתי הקבוצות בעץ הקבוצות. במידה ולפחות אחד מהם אינו נמצא , נחזיר FAILURE*.  
נמחק את שתי הקבוצות מהעץ לפי יכולת, ואת הקבוצה שנקנתה מהעץ לפי מזהה. לאחר מכן נבצע איחוד לפי גודל. נעדכן את פרטי הקבוצה בהתאם לאיחוד ונוסיף את הקבוצה המאוחדת לעץ לפי יכולת.  
בעת האיחוד לפי גודל נשתמש בשיטת הארגזים לשמירה על נכונות השדות :*

*מקרה א' : הקבוצה שקונה גדולה שווה מהקבוצה שנקנתה*

*מקרה ב' : הקבוצה שקונה קטנה מהקבוצה שנקנתה*

*סיבוכיות זמן :*

* *מציאת שתי הקבוצות בעץ הקבוצות-*
* *ביצוע איחוד לפי גודל-* משוערך*.*
* *עדכון נתונים -*

*סה''כ :* משוערך *.*