מבנה הנתונים יכיל:

1. - מונה של כל השחקנים במבנה הנתונים.
2. - מבנה מסוג Union Find שכל איבר בו הוא מספר קבוצת משחק, וכל קבוצה בו מייצגת קבוצות משחק שאוחדו לכדי קבוצה אחת.
3. – מערך בגודל של קבוצות המשחק, כאשר באינדקס ה- נשמור מידע אודות קבוצת השחקנים שמספרה .  
   כל קבוצה פעילה במשחק (קבוצה שלתוכה איחדנו קבוצות משחק נוספות) תכיל את המידע הבא:
   1. - מונה של מספר השחקנים בקבוצה.
   2. - עץ דרגות מסוג AVL שכל צומת בו מכיל ערך מסוים עבורו קיים לפחות שחקן אחד בקבוצה הנ"ל שערך ה- שלו זהה לערך הצומת.  
      העץ ממוין לפי המפתח ושומר בכל צומת את המידע הבא:

* *– מספר השחקנים בקבוצה שערך ה- שלהם שווה למפתח הצומת הנוכחי.*
* *– מספר השחקנים בקבוצה שערך ה- שלהם קטן או שווה למפתח הצומת הנוכחי.*
* *– סכום הניקוד של כל השחקנים בקבוצה שערך ה- שלהם קטן או שווה למפתח הצומת הנוכחי.*
  1. - מערך בגודל שכל צומת בו מצביע לעץ דרגות מסוג AVL ששמו- שכל צומת בו מכיל ערך מסוים עבורו קיים לפחות שחקן אחד בקבוצה הנ"ל שערך ה- שלו זהה לערך הצומת וכן שערך ה- שלו זהה לאינדקס המתאים במערך .

תוכן עץ זה זהה לתוכן העץ פרט לעובדה שבעץ זה מוחזק מידע אודות שחקנים בקבוצה שערך ה- שלהם זהה לאינדקס המתאים במערך .

1. - עץ דרגות מסוג AVL שכל צומת בו מכיל ערך מסוים עבורו קיים לפחות שחקן אחד במשחק כולו שערך ה- שלו זהה לערך הצומת.  
   תוכן עץ זה זהה לתוכן העץ פרט לעובדה שבעץ זה מוחזק מידע אודות כל השחקנים בקבוצה שערך ה- שלהם חיובי ממש.
2. - מערך בגודל שכל צומת בו מצביע לעץ דרגות מסוג AVL ששמו- שכל צומת בו מכיל ערך מסוים עבורו קיים לפחות שחקן אחד במשחק שערך ה- שלו זהה לערך הצומת וכן שערך ה- שלו זהה לאינדקס המתאים במערך .  
   תוכן עץ זה זהה לתוכן העץ פרט לעובדה שבעץ זה מוחזק מידע אודות שחקנים במשחק שערך ה- שלהם זהה לאינדקס המתאים במערך .
3. – טבלת ערבול של כלל השחקנים במערכת (תרחיבי פה באיזה פתרון התנגשות בחרת- כנראה ש- tree hashing) שכל איבר בה הוא אובייקט .

מחלקות

1. - מחלקה לשמירת מידע אודות שחקן במבנה. מכילה את המידע:
   1. – מזהה שחקן.
   2. - מזהה הקבוצה שהשתייך אליה השחקן, כפי שהיה נתון על ידי המשתמש בהכניסו למבנה הנתונים.
2. – מחלקה לשמירת מידע אודות קבוצה במבנה. מכילה את המידע:

מימושים לפונקציות:

* 1. הקצאת מערך בגודל *ואתחול כל תא בערך המתאים לאינדקס   
     שלו בתוספת 1.*
  2. הקצאת מערך בגודל *ואתחול התאים ב- NULL.*
  3. *אתחול .*
  4. *לכל נקצה אובייקט חדש של מחלקת בעל השדות הבאים:*
* *ניצור עץ AVL ריק תחת השדה .*
* *נקצה מערך* בגודל תחת השדה *ונאתחל את התאים   
  ב- NULL.*
  1. *נקצה מבנה* מסוג Union Find*בשם .*
  2. *לכל נקרא לפונקציה של מבנה .*

*סיבוכיות*

* 1. *הקצאת מערך בגודל ואתחולו- .*
  2. *הקצאת מערך בגודל ואתחולו כאשר זהו ערך קבוע מבחינת שיקולי חישובים- .*
  3. *אתחול משתנה- .*
  4. *לכל נעשה מספר קבוע של פעולות (נזכור ש- של עץ AVL כפי שמימשנו בתרגיל רטוב 1 היא בסיבוכיות וכן הקצאת מערך בגודל ואתחולו כאשר זהו ערך קבוע מבחינת שיקולי חישובים נעשה ב- ) לכן בסה"כ .*
  5. *קריאה ל- של מבנה Union Find- סיבוכיות .*
  6. *לכל נעשה פעולה בסיבוכיות זמן כפי קראינו בתרגול ולכן בסה"כ .*

*לכן בסה"כ נקבל סיבוכיות זמן .*

* 1. *אם* 𝐺𝑟𝑜𝑢𝑝𝐼𝐷 ≤ 0, 𝐺𝑟𝑜𝑢𝑝𝐼𝐷 > 𝑘 ,DS==NULL או 𝑠𝑐𝑜𝑟𝑒 ≤ 0, 𝑠𝑐𝑜𝑟𝑒 > 𝑠𝑐𝑎𝑙𝑒   
     או ,𝑃𝑙𝑎𝑦𝑒𝑟𝐼𝐷 ≤ 0 נחזיר שגיאה.
  2. *נחפש את השחקן שהמזהה שלו הוא בטבלת הערבול . אם הוא נמצא במבנה- נחזיר שגיאה.*
  3. *אחרת:*
* *נייצר אובייקט מנתוני השחקן ונכניסו לטבלת הערבול על ידי שימוש במפתח .*
* *נגדיל ב-1 את .*
* *נבצע ב- ונמצא את המספר העדכני של הקבוצה אליה נרצה להוסיף את השחקן. ניגש לאיבר המתאים במערך (אינדקס ששווה למספר הקבוצה שקיבלנו מפונקציית החיפוש מינוס 1).*
* *נעבור בעזרת המצביע הנמצא באיבר המתאים שמצאנו בשלב הקודם לאובייקט המתאים, ונגדיל את הערך שבשדה ב-1.*

*סיבוכיות*

* 1. *מספר קבוע של פעולות- .*
  2. *חיפוש בטבלת ערבול בגודל - בממוצע על הקלט.*
* *הכנסה לטבלת ערבול בגודל - בממוצע על הקלט.*
* *מספר קבוע של פעולות - .*
* *התייחסות לסיבוכיות משוערכת בסוף.*
  1. *לכל נעשה מספר קבוע של פעולות (נזכור ש- של עץ AVL כפי שמימשנו בתרגיל רטוב 1 היא בסיבוכיות וכן הקצאת מערך בגודל ואתחולו כאשר זהו ערך קבוע מבחינת שיקולי חישובים נעשה ב- ) לכן בסה"כ .*
  2. *קריאה ל- של מבנה Union Find- סיבוכיות .*
  3. *לכל נעשה פעולה בסיבוכיות זמן כפי קראינו בתרגול ולכן בסה"כ .*

*לכן בסה"כ נקבל סיבוכיות זמן .*

* 1. *אם DS==NULL או או נחזיר שגיאה.*
  2. *נחפש את השחקן שהמזהה שלו הוא בטבלת הערבול . אם   
     הוא לא נמצא במבנה- נחזיר שגיאה.*
  3. *אחרת, נשמור את שדות המחלקה של אובייקט שמצאנו בטבלת הערבול.*
  4. *נבצע ב- ונמצא את המספר העדכני של הקבוצה אליה השחקן שייך. ניגש לאיבר המתאים במערך (אינדקס ששווה למספר הקבוצה שקיבלנו מפונקציית החיפוש מינוס 1).*
  5. *נבצע חיפוש בעץ לפי מפתח ששמרנו בסעיף הקודם, ונקטין ב-1 את ערך השדה . אם לאחר העדכון נסיר את הצומת בעלת המפתח מהעץ .*
  6. *נבצע חיפוש בעץ לפי מפתח העדכני- ערך זה שווה לערך המפתח ששמרנו בסעיף 5.4 בתוספת הערך הנתון ב- .   
     אם צומת זה קיים בעץ- נגדיל ב-1 את ערך השדה .   
     אחרת, נוסיף לעץ את הצומת שמפתחה הוא ערך העדכני, עם .*
  7. *ניגש לאינדקס ששמרנו בשלב 5.4 המתאים ל- של השחקן הנ"ל, במערך . כעת נחזור על שלבים 5.5 ו- 5.6 הפעם על העץ המתאים.*
  8. *נחזור על שלבים 5.5 ו-5.6 הפעם על העץ .*

*סיבוכיות*

* 1. *מספר קבוע של פעולות- .*
  2. *חיפוש בטבלת ערבול בגודל - בממוצע על הקלט.*
  3. *מספר קבוע של פעולות- .*
  4. *התייחסות לסיבוכיות משוערכת בסוף*
  5. *חיפוש ומחיקה בעץ AVL בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  6. *חיפוש והוספה לעץ AVL בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק  
     כולו) כפי שנלמד בהרצאה .*
  7. *גישה לאיבר במערך נעשית ב- ואותם נימוקים של הסעיפים הקודמים תקפים לפעולות בסעיף זה שכן המבנים עליהם הפעולות נעשות הם זהים ומכילים לכל יותר את אותה הכמות של איברים.*
  8. *אותם נימוקים מהסעיפים הקודמים תקפים לפעולות הנעשות בסעיף זה שכן המבנים עליהם הפעולות נעשות הם זהים ומכילים כמות איברים ו- כאשר אלה הם מספרי השחקנים והקבוצות במשחק כולו.*

*לכן בסה"כ נקבל סיבוכיות זמן .*

* 1. *אם אחד המצביעים שווה ל- NULL או , או נחזיר שגיאה.*
* *אם* 
  1. *נבצע ב- ונמצא את המספר העדכני של הקבוצה אליה השחקן שייך. ניגש לאיבר המתאים במערך (אינדקס ששווה למספר הקבוצה שקיבלנו מפונקציית החיפוש מינוס 1).*
  2. *אם נחזיר שגיאה.*
  3. *נחלץ את ערך של הצומת בעל המפתח המקסימלי בעץ . נסמן ערך זה ב- ואת המפתח הנ"ל ב- .*
  4. *נמצא את הצומת בעץ עבורו הוא בעל הערך המינימלי שגדול או שווה ל- . נעשה זאת על ידי שימוש בפונקציה . נסמן ערך זה ב- ואת המפתח הנ"ל נסמן ב- .  
       
       
     אם   
     נבחין כי כל יתר השחקנים בקבוצה שאינם נספרים בדרגות העץ הם שחקנים בעלי ולכן נחזיר את הערך:*

*כאשר הינו ערך של הצומת בעץ .*

*אחרת,   
נבחין שמתקיים   
ולכן נחזיר את הערך:*

*כאשר:  
 הינו ערך של הצומת בעץ ) הינו ערך של הצומת .  
 הינו ערך של הצומת .*

* *אם   
  נחזור על שלבים 8.3, 8.4, 8.5 רק הפעם עבור עץ הכללי, והערך*

*צריכות להוסיף מידע על מבנה הנתונים  
 union find*

*hash table*

*ranked tree*

*ועל הפונקציות המסופקות על ידי כל אחד מהם עם הסברי סיבוכיות של "ראינו בהרצאה".*