נממש 3 מבני נתוני שבהם נשתמש במבנה שלנו:

1. **Union Find** – נממש מבנה זה בדומה לנלמד בהרצאה עם עצים הפוכים וכיווצים. לשם כך נשתמש ב-2 מערכים: מערך size המחזיק עבור קבוצות קיימות את גודלן באינדקס המתאים, ומערך parent המייצג עץ הפוך כך שעבור שורש מוחזק האינדקס עצמו, ועבור כל צומת אחר, מוחזק האינדקס לערך הצומת אליו הוא מצביע- בדיוק כפי שראינו בהרצאה. מבנה נתונים זה יכיל את הפעולות:
   1. **Find()** – כפי שנלמד בהרצאה סיבוכיות משוערכת יחד עם Union .
   2. **Union()** - כפי שנלמד בהרצאה סיבוכיות משוערכת יחד עם Find .
   3. **MakeSet(i)** –מחזיר קבוצה חדשה בעלת איבר בודד .
2. **עץ AVL דרגות** – לשם מימוש זה נשתמש במימוש לעץ AVL מהרטוב הקודם, כאשר לכל צומת בעץ נוסיף את השדות הבאים:  
   **w\_info –** מחלקה גנרית המכילה את כל המידע הנוסף שנרצה לשמור עבור כל צומת בעץ.  
   **w –** אותה מחלקת מידע נוסף שאנו שומרים עבור כל צומת בעץ, המכילה את הערך המצטבר משמע את סכום הערכים של המידע הנוסף של תת העץ של הצומת הנוכחי (כולל ערכיו של הצומת הנוכחי).  
   סיבוכיות: כפי שראינו בהרצאה, ניתן לתחזק מידע נוסף בו בכל צומת נשמר מידע על תת העץ שמתחתיו בעזרת עדכונים מתאימים בזמן הכנסות, הסרות וגלגולים, באופן שלא ייפגע בסיבוכיות של כל אחת מפעולות אלה.  
   כמו כן נוסיף למבנה העץ את הפונקציות הבאות:  
   **Rank()-** חישוב דרגה של המידע הנוסף אותו אנו שומרים עבור כל צומת.  
   סיבוכיות: כפי שראינו בהרצאה, עבור עץ AVL סיבוכיות הזמן היא .

**SelectMinBigger(k)-** בחירת צומת בעל הדרגה המינימלית שגדולה או שווה ל-k, כאשר הדרגה מחושבת ביחס למידע הנוסף אותו אנו שומרים בכל צומת. (כלומר k הינו אובייקט של המחלקה הגנרית לשמירת מידע נוסף, ונרצה למצוא את הצומת בעץ שסכום המידעים הנוספים של כל הצמתים שקטנים או שווים לו, הוא המינימלי שגדול או שווה ל-k).  
סיבוכיות: הפעולה הנ"ל פועלת לפי האלגוריתם שהוצג בתרגול למימוש Select פאת לשינויים הבאים:  
- בכל בדיקה של ערך במקום לבצע השוואה ל- נבצע השוואה ל-

- תיאור אלגוריתם  
3) **טבלת ערבול דינאמית** – נממש טבלת ערבול עם chain hashing. טבלת הערבול תהיה דינאמית כפי שנלמד בתרגול, כך שגודלה יקבע בהתאם למספר הערכים. לטבלה קיימות הפעולות הבאות: (נתייחס לסיבוכיות כשגודל הטבלה קבוע, ובסוף נתייחס לסיבוכיות הנוספת משינויי הגודל).

**Find() –** מוצאת את הערך אם הוא נמצא בטבלה, מחזירה אם הוא לא נמצא. כפי שנלמד בהרצאה סיבוכיות בממוצע על הקלט.

**Insert() –** מכניסה ערך חדש לטבלה, אם הוא כבר קיים היא מחזירה זאת. כפי שנלמד בהרצאה סיבוכיות בממוצע על הקלט.

**Remove() –** מסירה ערך מהטבלה, אם הוא לא נמצא מחזירה זאת. כפי שנלמד בהרצאה סיבוכיות בממוצע על הקלט.

* **סיבוכיות מערך דינאמי -** כפי שהוכח בתרגול הסיבוכיות המשוערכת של פעולות הכנסה והוצאה על מערך דינאמי היא . לכן כל הפעולות של מבנה זה הם בסבוכיות משוערכת בממוצע על הקלט.

מבנה הנתונים יכיל:

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

* *– שלב במשחק.*
* *– מידע נוסף אודות מספר השחקנים ברמה ) וכן את המכפלה של מספר שחקנים זה במספר השלב  
  .*
* *– ערך מצטבר של המידע הנוסף כפי שפורט לעיל.*
  1. - מערך בגודל שכל צומת בו מכיל:
* 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.

מבנה העץ זהה למבנה העץ .

* - מספר השחקנים בקבוצה בעלי ניקוד זה.

1. - עץ דרגות מסוג AVL שכל צומת בו מכיל ערך עבורו קיים לפחות שחקן אחד במשחק שערך ה- שלו זהה לערך הצומת.  
   מבנה עץ זה זהה למבנה העץ .
2. - מערך בגודל שכל צומת בו מכיל:

* 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 .  
  מבנה עץ זה זהה למבנה העץ .
* - מספר השחקנים במשחק בעלי ניקוד זה.

1. – טבלת ערבול של כלל השחקנים במערכת 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:
3. – מחלקה לשמירת המידע הנוסף עבור כל צומת. מכילה את המידע:
   1. - מספר השחקנים באותו שלב
   2. - מספר השחקנים באותו שלב, מוכפל במספר השלב.

ציור להמחשת מבנה הנתונים:

**Players**

טבלת ערבול דינאמית

All player counter

UF

של groups

**Groups array**

playerID

groupID

level

score

Group player counter

Group level

Group scores array

Level

Num of players

Rank count level

Rank sum level

Group player score counter

Group score levels

Level

Num of players

Rank count level

Rank sum level

**Scores array**

players score counter

score levels

Level

Num of players

Rank count level

Rank sum level

Level

Num of players

Rank count level

Rank sum level

**levels**

|  |
| --- |
|  |
|  |
|  |
|  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

מימושים לפונקציות: נסמן מספר הקבוצות-k, מספר השחקנים-n.

* 1. הקצאת מערך בגודל *ואתחול התאים ב- NULL.*
  2. הקצאת מערך *בגודל ואתחול התאים באובייקט דיפולטי.*
  3. *אתחול .*
  4. *נקצה מבנה* מסוג Union Find*בשם .*
  5. *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. *קריאה ל- של מבנה Union Find- סיבוכיות .*
  5. *לכל נעשה פעולה בסיבוכיות זמן כפי שראינו בתרגול ולכן בסה"כ .*

*לכן בסה"כ נקבל סיבוכיות זמן .*

1. **–**
   1. אם DS==NULL, ,, או נחזיר שגיאה.
   2. *נבצע , ב- ונמצא את מזהה הקבוצה שלהן. נסמנם* , בהתאמה.
   3. אם מזהי הקבוצות זהות הם כבר באותה קבוצה ולכן סיימנו.
   4. נבצע ונשמור *את מזהה הקבוצה החדשה, זוהי הקבוצה אליה נמזג את הקבוצה השנייה, בה"כ .*
   5. נמזג את נתוני הקבוצה אל הקבוצה *. מיזוג הקבוצות יכלול:*
      * *איחוד עצי הדרגות של 2 הקבוצות והשמת העץ החדש בקבוצה .*
      * *מעבר על מערך של 2 הקבוצות, עבור כל אינדקס (המסמן ניקוד של השחקנים) נסכום את מספר השחקנים עבור ניקוד זה ב-2 המערכים. בנוסף:*
        + *אם אין עצים בתא בעל אינדקס או אם יש עץ בתא בעל אינדקס זה רק ב- זה- נמשיך.*
        + *אם יש עץ בתא בעל אינדקס זה רק ב- נעתיק אותו למערך של הקבוצה בתא בעל האינדקס הנוכחי.*
        + *אם יש עץ ב-2 המערכים בתא זה, נמזג אותם ונכניס את העץ החדש לתא הנכון במערך של הקבוצה .*

סיבוכיות

* 1. מספר *קבוע של פעולות- .*
  2. *התייחסות לסיבוכיות משוערכת בסוף.*
  3. מספר *קבוע של פעולות- .*
  4. *התייחסות לסיבוכיות משוערכת בסוף.*
  5. \* איחוד 2 עצים כאשר הגודל שלהם ביחד הוא לכל היותר כמספר השחקנים במבנה , לכל היותר .  
     \* מעבר על מערך בגודל קבוע וביצוע מספר קבוע של פעולות (נתייחס בנפרד לפעולת מיזוג העצים).  
     \* מספר הצמתים בכל העצים ב-2 המערכים הוא לכל היותר כמספר השחקנים (כי כל צומת מייצגת רמה של שחקן בניקוד מסוים בקבוצה מסוימת) ולכן מיזוג כל העצים סה"כ הוא לכל היותר .

*לכן בסה"כ נקבל סיבוכיות זמן* משוערך.

* 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 מזהה הקבוצה אליה נוסיף את השחקן. ניגש לאיבר המתאים במערך .*
* *נעבור בעזרת המצביע הנמצא באיבר המתאים שמצאנו בשלב הקודם לאובייקט , ואם לא קיימת שם קבוצה עדיין- ניצור קבוצה .*
* *נגדיל את הערך שבשדה ב-1.*
* *ניגש למערך – הנמצא באובייקט זה לאינדקס ונגדיל גם שם את ב-1.*

*סיבוכיות*

* 1. *מספר קבוע של פעולות- .*
  2. *חיפוש בטבלת ערבול - משוערך בממוצע על הקלט.*
* *הכנסה לטבלת ערבול - משוערך בממוצע על הקלט.*
* *מספר קבוע של פעולות - .*
* *התייחסות לסיבוכיות משוערכת בסוף.*
* *אתחול קבוצה חדשה אתחול של משתנים בגודל קבוע סיבוכיות .*
* *מספר קבוע של פעולות - .*

*לכן בסה"כ נקבל סיבוכיות זמן משוערך בממוצע על הקלט.*

1. **–**
   1. אם DS==NULL ,או 𝑃𝑙𝑎𝑦𝑒𝑟𝐼𝐷 ≤ 0 נחזיר שגיאה.
   2. נבדוק את קיומו של השחקן בעל בטבלת הערבול אם הוא לא קיים נחזיר שגיאה.
   3. אם הוא קיים נשמור את שלו.
   4. *נבצע ב- ונמצא את מזהה הקבוצה ממנה נרצה להסיר את השחקן. ניגש לאיבר המתאים במערך .*
   5. נקטין את מונה מספר השחקנים בקבוצה ב-1.
   6. ניגש לתא בעל האינדקס במערך נקטין שם את מספר השחקנים בעלי ניקוד זה ב-1, נסיר מהעץ שבתא זה את *הצומת שמפתחו*  (מסעיף 4.3) ולבסוף *נוסיף לעץ צומת חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש* )כדי לא לפגוע בנכונות המבנה אנו מוציאים ומכניסים צומת ולא ניגשים למשתנה ומשנים אותו).
   7. *נסיר מהעץ את הצומת שמפתחו , ונוסיף לעץ צומת חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש.*
   8. נמחק את השחקן מטבלת הערבול .
   9. נקטין את מונה מספר השחקנים במבנה הנתונים ב-1.
   10. ניגש לתא בעל האינדקס במערך נקטין שם את מספר השחקנים בעלי ניקוד זה ב-1, נסיר מהעץ שבתא זה את *הצומת שמפתחו*  ולבסוף *נוסיף לעץ צומת חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש.*
   11. *נסיר מהעץ את הצומת שמפתחו , ונוסיף לעץ צומת חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש.*

סיבוכיות

* 1. מספר *קבוע של פעולות- .*
  2. *חיפוש בטבלת ערבול - משוערך בממוצע על הקלט.*
  3. מספר *קבוע של פעולות- .*
  4. *התייחסות לסיבוכיות משוערכת בסוף*
  5. מספר *קבוע של פעולות- .*
  6. *חיפוש, הסרה והוספה בעץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  7. *חיפוש, הסרה והוספה בעץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  8. *חיפוש והסרה בטבלת ערבול בגודל - בממוצע על הקלט.*
  9. מספר *קבוע של פעולות- .*
  10. *חיפוש, הסרה והוספה בעץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  11. *חיפוש, הסרה והוספה בעץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*

*לכן בסה"כ נקבל סיבוכיות זמן* משוערך, בממוצע על הקלט.

* 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. *נסיר מהעץ את הצומת שמפתחו (ששמרנו בסעיף הקודם), ונוסיף לעץ את צמות חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש.*
  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- נסירו מהעץ ונוסיף צומת בעל אותו מפתח וערך הגדול ב-1 מהערך הקודם.  
     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 .*
  9. *נחזור על שלב 5.7 על .*

*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.*
  9. *כנ"ל.*

*לכן בסה"כ נקבל סיבוכיות זמן משוערך, בממוצע על הקלט.*

1. ***–***
   1. אם DS==NULL, 𝑃𝑙𝑎𝑦𝑒𝑟𝐼𝐷 ≤ 0, , או נחזיר שגיאה.
   2. נבדוק את קיומו של השחקן בעל בטבלת הערבול אם הוא לא קיים נחזיר שגיאה.
   3. אם הוא קיים נשמור את שלו.
   4. *נבצע ב- ונמצא את המספר העדכני של הקבוצה בא נמצא השחקן. ניגש לאיבר המתאים במערך (אינדקס ששווה למספר הקבוצה שקיבלנו מפונקציית החיפוש מינוס 1).*
   5. ניגש לתא בעל האינדקס במערך , נקטין שם את מספר השחקנים בעלי ניקוד זה ב-1.
   6. *נסיר מהעץ את הצומת שמפתחו ששמרנו ב6.3, ונוסיף לעץ צומת חדש בעל מפתח וערך הקטן ב-1 מהערך הקודם, כל זאת וערך זה חיובי ממש.*
   7. ניגש לתא בעל האינדקס במערך , נגדיל שם את מספר השחקנים בעלי ניקוד זה ב-1.
   8. נבצע חיפוש על העץ שבתא זה במערך, לפי המפתח של השחקן. *אם צומת זה קיים בעץ- נסיר מהעץ את הצומת שמפתחו ונוסיף לעץ צומת חדש בעל מפתח וערך הגדול ב-1 מהערך הקודם, אחרת נוסיף לעץ צומת שמפתחו ערך .*
   9. נחזור על פעולות 6.5-6.8 על המערך .
   10. נחפש את השחקן בעל בטבלת הערבול ונשנה את

ל-.

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. מספר *קבוע של פעולות- .*
  6. *חיפוש, הסרה והוספה בעץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  7. מספר *קבוע של פעולות- .*
  8. *חיפוש, הסרה והוספה עץ AVL דרגות בגודל לכל היותר (לא ייתכנו יותר רמות משחקנים במשחק כולו) כפי שנלמד בהרצאה .*
  9. *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.*
  10. *חיפוש בטבלת ערבול בגודל - בממוצע על הקלט ו*מספר *קבוע של פעולות- .*

*לכן בסה"כ נקבל סיבוכיות זמן* משוערך, בממוצע על הקלט.

* 1. אם DS==NULL, או נחזיר שגיאה.
  2. עבור :
     + *נבצע ב- ונמצא את המספר העדכני של הקבוצה בא נמצא השחקן. ניגש לאיבר המתאים במערך (אינדקס ששווה למספר הקבוצה שקיבלנו מפונקציית החיפוש מינוס 1).*
     + חישוב המונה:
       - ניגש לתא בעל האינדקס במערך .

עתה נרצה לחשב את מספר השחקנים בעץ בתחום הרמות המבוקש:

* + - * במקרה ובו ערך הצומת המקסימלי בעץ קטן מ- נחזיר 0, אם הוא שווה לו נחזיר את מספר השחקנים בשלב זה.
      * נסמן את מספר השחקנים שרמתם קטנה שווה ל-, .

ואת מספר השחקנים שרמתם קטנה שווה ל-, .

* + - * נשתמש בפונקציה של עץ הדרגות שלנו על מנת למצא את הצומת המינימלי שרמתו גדולה מ- נסמנה .
      * בעזרת על נמצא את מספר השחקנים בעץ שרמתם קטנה

מ- כלומר קטנה שווה ל- , נכניס ערך זה ל-.

* + - * אם הרמה המקסימלית בעץ קטנה שווה ל-, נבצע על הצומת המקסימלי בעץ. ונכניס את מספר השחקנים הקטן שווה לצומת זה ל-.
      * אחרת, נשתמש בפונקציה על מנת לקבל את הצומת המינימלית הגדולה ביותר מ- נסמנה .
      * בעזרת על נמצא את מספר השחקנים בעץ שרמתם קטנה

מ- כלומר קטנה שווה ל-, נכניס ערך זה ל-.

* + - * עתה נחשב את מספר השחקנים בעלי רמה (אם קיימים) נסמנם :
        1. אם – נבדוק אם הוא נמצא בעץ ואם כן נשמור את מספר השחקנים ברמה זו ב-.
        2. אם – נחשב את מספר השחקנים ברמה 0 בעזרת חיסור מספר השחקנים בעץ (מספר השחקנים שרמתם גדולה מ-0) ממספר השחקנים הכולל. נשמור את המספר ב- *.*

המונה יהיה:

*חישוב המכנה:*

* + - * *ניגש לעץ .*
      * *בצורה דומה נחשב מספר שחקנים בעץ בתחום רמות מבוקש.*
      * המכנה יהיה: מספר השחקנים בטווח אותו מצאנו.

*נחזיר את המונה חלקי המכנה.*

* 1. עבור :
     + נבצע פעולות זהות על המערך והעץ .

סיבוכיות

* 1. מספר *קבוע של פעולות- .*
  2. *התייחסות לסיבוכיות משוערכת בסוף*
     + *חישוב המונה:*
       - מספר *קבוע של פעולות- .*
       - חישוב דרגה בעץ דרגות בעל לכל היותר צמתים, כפי שפורט לעיל .
       - חישוב דרגה בעץ דרגות בעל לכל היותר צמתים, כפי שפורט לעיל .
       - חיפוש צומת בעץ בעל לכל היותר צמתים, .
       - מספר *קבוע של פעולות- .*
     + *חישוב המכנה:*
       - מספר *קבוע של פעולות- .*
       - חישוב דרגה בעץ דרגות בעל לכל היותר צמתים, כפי שפורט לעיל .
       - חישוב דרגה בעץ דרגות בעל לכל היותר צמתים, כפי שפורט לעיל .
       - חיפוש צומת בעץ בעל לכל היותר צמתים, .
       - מספר *קבוע של פעולות- .*
  3. *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* משוערך.

* 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*

*סיבוכיות*

1. ***-***
   1. *שחרור טבלת הערבול .*
   2. *מעבר על המערך כאשר בכל תא במערך אם יש בו מצביע לקבוצה ניגש אל הקבוצה:*
      * *נשחרר את העץ .*
      * *נעבור על המערך ובכל תא בו יש עץ נשחרר אותו.*
   3. *שחרור העץ .*
   4. *נעבור על המערך ובכל תא בו יש עץ נשחרר אותו.*
   5. *שחרור מבנה הנתונים groups.*

*סיבוכיות*

* 1. *מעבר על מערך בגודל לכל היותר , .*
  2. *מעבר על מערך בגודל סיבוכיות . נחשב את סיבוכיות הפעולות שנבצע על כל הקבוצות ביחד:*
     + *נסמן את מספר השחקנים בקבוצה , . נקבל כי .*

*נשים לב כי כמות הצמתים בעץ עבור קבוצה הוא לכל היותר כי יתכנו שחקנים בעלי רמה 0 שאינם בעץ. לכן נקבל ששחרור כל עצי עבור כל הקבוצות הוא בסיבוכיות לכל היותר:*

* + - *מעבר על מערך בגודל קבוע, סיבוכיות .*
    - *מספר הצמתים בכל העצים במערך עבור קבוצה הוא לכל היותר (כל צומת מייצגת שחקן בקבוצה שרמתו לא 0) לכן סה"כ עבור שחרור כל עצי עבור קבוצה הוא בסיבוכיות לכל היותר , ולכן שחרור כל העצים בכל הקבוצות הוא בסיבוכיות לכל היותר .*
  1. *שחרור עץ בגודל לכל היותר , סיבוכיות .*
  2. *מעבר על מערך בגודל קבוע, סיבוכיות , נתייחס לשחרור כל העצים ביחד. בצורה דומה לשחרור המערך בקבוצות נקבל שחרור כל עצי הוא לכל היותר בסיבוכיות .*
  3. *שחרור מבנה UF .*

*לכן בסה"כ נקבל סיבוכיות זמן .*

***סיבוכיות מקום של מבנה הנתונים***

1. – סיבוכיות מקום .
2. – מבנה UF בגודל , סיבוכיות מקום
3. – מערך בגודל סיבוכיות מקום .

נתייחס עבור מבני הקבוצה שסיבוכיות המקום שלהם אינה , למבנה הנתונים של כל הקבוצות סה"כ:

* 1. - כל צומת בעץ מייצג שלב בו יש שחקנים (שהוא אינו 0) לכן סה"כ מספר הצמתים בכל עצי הקבוצות הוא לכל היותר כמספר השחקנים במנה . לכן נקבל סיבוכיות של לכל היותר .
  2. - מערך בגודל קבוע סיבוכיות מקום . סה"כ מספר הצמתים בכל העצים במערך הינו לכל היותר מספר השחקנים בקבוצה. ולכן עבור כל הקבוצות נקבל שמספר הצמתים בכל המערכים הוא לכל היותר כמספר השחקנים במבנה, לכן נקבל סיבוכיות של לכל היותר .

1. - עץ שמספר הצמתים בו הוא לכל היותר כמו השחקנים במבנה, לכל היותר .
2. – מערך בגודל קבוע סיבוכיות מקום . סה"כ מספר הצמתים בכל העצים במערך הינו לכל היותר מספר השחקנים במבנה ולכן נקבל סה"כ סיבוכיות של לכל היותר .
3. – גודל המערך הדינאמי הוא לכל היותר כי אחרת נגדיל את המערך ולכן סיבוכיות המקום של המערך הוא . בנוסף מספר הצמתים (של רשימות מקושרות) הוא כמספר השחקנים במבנה ולכן סה"כ נקבל סיבוכיות מקום של .

*לכן בסה"כ נקבל סיבוכיות מקום .*

*צריכות להוסיף מידע על מבנה הנתונים  
 union find*

*ranked tree*

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

*סיבוכיות של 8*