



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 1 מתוך 8

רעיון כללי

מבנה הנתונים שלנו יחזיק טבלת ערבול של שחקנים ו-UnionFind של קבוצות. כל קבוצה תכיל מערך בגודל **scale** של מבנה חדש שבנינו- **score_structure**, בו כל תא מייצג תוצאה ספציפית. כל תא כזה מחזיק בתוכו עץ **AVL** בו כל השחקנים בעלי רמה שונה מאפס מאותה תוצאה ממוינים לפי רמה, ומשתנה שיחזיק את מספר השחקנים ברמה אפס מאותה תוצאה. כל העצים במערכת הם עצי **AVL** בעלי מידע נוסף: **rank1**- מספר הצמתים בתת העץ ו**rank2**- סכום הרמות בתת העץ. בנוסף, הצמתים בכל העצים שלנו מכילים מצביעים לשחקנים (**class PlayerSeat**), על מנת לחסוך במקום בזיכרון. הצמתים בכל העצים ממוינים לפי רמה בחשיבות ראשונה, ותעודת זהות בחשיבות שניה. בכל קבוצה, התא הראשון במערך (המיוצג על ידי התוצאה אפס) יכיל את כל המידע הנ"ל תקף לכל השחקנים מאותה הקבוצה, כלומר בלי חשיבות לתוצאה. בנוסף, הקבוצה בעלת המזהה 0, תכיל את אותם נתונים, בהסתכלות על כלל המערכת (כלומר כל השחקנים מכל הקבוצות).

פירוט המידע של מבנה הנתונים שלנו:

k- מספר הקבוצות במערכת.

scale- התוצאה המקסימלית במערכת.

PlayersTable- טבלת ערבול של שחקנים, ממומשת באמצעות מערך דינאמי ופונקציית ערבול של מודולו גודל המערך. טבלת הערבול שומרת על פקטור עומס קטן מ-2 לכל אורך הריצה, בדומה לנלמד בהרצאה (צמצום או הגדלת המערך בהכנסה והוצאה).

כל איבר בטבלה (**class Player**) יכיל את המידע הבא:

- **playerID**- תעודת הזהות של השחקן.
- **groupID**- מספר הקבוצה של השחקן.
- **level**- רמת השחקן.
- **score**- תוצאת השחקן.

GroupsUF- מבנה UnionFind של קבוצות.

המבנה יחזיק שלושה מערכים בגודל **k+1**:

- **k**- מספר הקבוצות במערכת.
- **size**- מערך של מספרים שלמים אשר מחזיק בתא ה-**i** את מספר הקבוצות בתוך הקבוצה ה-**i**.
- **parent**- מערך של מספרים שלמים אשר מחזיק בתא ה-**i** את ה"אבא" של הקבוצה ה-**i**.
- **groups**- מערך של קבוצות בגודל **k+1**, כאשר הקבוצה 0 תחזיק את הנתונים עבור כלל המערכת.

כל קבוצה (**class Group**) תחזיק את המידע הבא:

- **groupID**- מספר הקבוצה
- **size**- מספר השחקנים בקבוצה.
- **scale**- טווח התוצאות במערכת.
- **groupSS**- מערך של **score_structure** בגודל **scale+1**

כל איבר במערך (**score structure**) יכיל את המידע הבא:

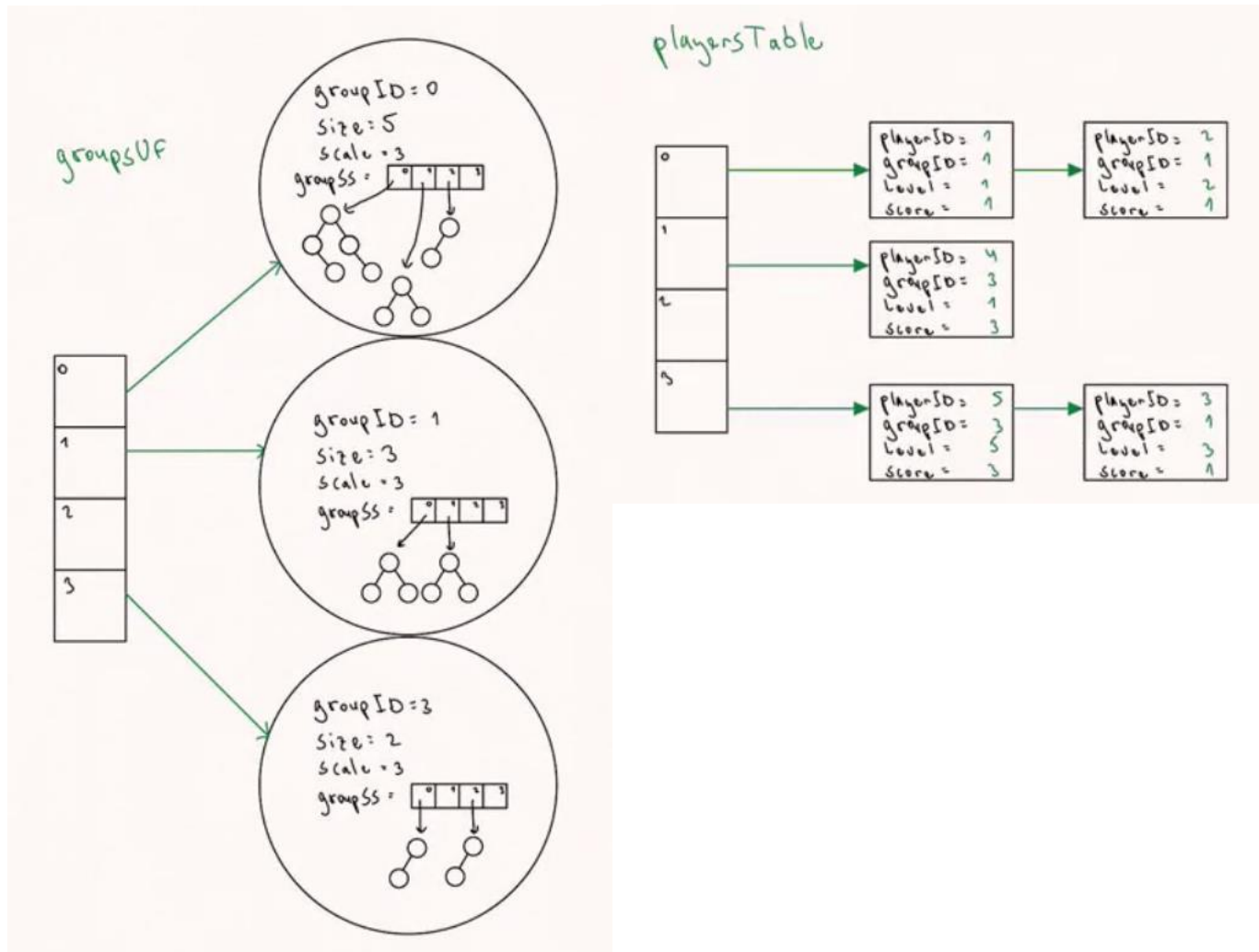


מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 2 מתוך 8

- **score** - התוצאה אותה התא מייצג, מספר בין 0 ל-**scale** כאשר התא 0 מתייחס לכל השחקנים ללא חשיבות לתוצאה שלהם באותה קבוצה.
- **zero_counter** - מספר השחקנים בתוצאה הספציפית בעלי רמה 0.
- **score_tree** - עץ AVL של מצביעים לשחקנים, בו נמצאים כל השחקנים בתוצאה הספציפית שרמתם גדולה מ-0.

שרטוט המבנה





מימוש הפונקציות

הערה לחישוב סיבוכיות הזמן בכלל הפונקציות: הסיבוכיות המשוערכת של כל הפונקציות תוסבר לאחר תיאור כל הפונקציות.

void* init(int k, int scale)

נבצע את הפעולות הבאות:

- נבדוק את תקינות הקלט, במידה ולא תקין נחזיר NULL.
- נאתחל את מספר הקבוצות והתוצאה המקסימלית בערכים שקיבלנו.
- נאתחל את טבלת הערבול לטבלה מינימלית ריקה בגודל 2.
- נקצה דינאמית את שלושת המערכים הנדרשים ל-UnionFind.
- נעבור בלולאה על כל הקבוצות, בכל קבוצה נאתחל את מספר הקבוצה להיות כנדרש.
- בנוסף בכל קבוצה נקצה דינאמית מערך בגודל $scale+1$, נעבור בלולאה על כל המערך ונאתחל את מספר התא להיות מספר התוצאה אותו הוא מייצג.

סיבוכיות זמן:

- אתחול טבלת הערבול – סיבוכיות זמן של $O(1)$.
- אתחול שלושת המערכים ב-UnionFind.
- מעבר בלולאה על כל מערכי ה-score_structure בכל הקבוצות- מכיוון ש-scale הינו לכל היותר 200, סיבוכיות זמן של $O(k \cdot scale) = O(k)$.
- כל שאר הפעולות- $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(k)$ במקרה הגרוע.

StatusType mergeGroups(void *DS, int GroupID1, int GroupID2)

נבצע את הפעולות הבאות:

- נבדוק את תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- נבדוק האם מספרי הקבוצות שווים, במידה וכן נחזיר הצלחה.
- נמצא את שורשי שתי הקבוצות, במידה וזהים נחזיר הצלחה, אחרת, נאחד אותם לפי גודל.
- איחוד הקבוצות יתבצע כך: נעבור בלולאה על כל מערך התוצאות, נבצע איחוד עצי AVL לפי האלגוריתם הנלמד בתרגול, בנוסף נעדכן את מוני שחקני רמות האפס בהתאם.

סיבוכיות זמן:

- מציאת שורשי שתי הקבוצות- מכיוון שמימשנו באמצעות קיצור מסלולים ואיחוד לפי גודל- סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- איחוד הקבוצות- איחוד עצים פועל בסיבוכיות זמן של $O(n_1+n_2)$ כאשר n_1 ו- n_2 הם מספר השחקנים בעצים, אנו מאחדים scale עצים כך שאם מספר השחקנים בשתי הקבוצות הינו n נקבל סיבוכיות זמן של $O(n)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + n)$ משוערך, בממוצע על הקלט.



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 – נתיב מאור 319002911, בר בן אבי 318176138
עמוד 4 מתוך 8

StatusType addPlayer(void *DS, int PlayerID, int GroupID, int score)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- נחפש את השחקן בטבלת הערבול- במידה ונמצא נחזיר שגיאה.
- במידה ולא נמצא, נכניס אותו לטבלת הערבול.
- נמצא את הקבוצה של השחקן ב-UF.
- נגדיל ב-1 את מונה שחקני הרמה 0 בתא במערך המיוצג על ידי תוצאת השחקן.
- נגדיל ב-1 את מונה שחקני הרמה 0 בתא 0, שמייצג את כל השחקנים בקבוצה מכל התוצאות.
- נחפש את הקבוצה 0, הקבוצה שמייצגת את כל השחקנים מכל הקבוצות.
- נגדיל ב-1 את מונה שחקני הרמה 0 בתא במערך המיוצג על ידי תוצאת השחקן מכל הקבוצות.
- נגדיל ב-1 את מונה שחקני הרמה 0 בתא 0 שמייצג את כל השחקנים מכל הקבוצות ומכל התוצאות.

סיבוכיות זמן:

- חיפוש השחקן בטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- הכנסת השחקן לטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- חיפוש הקבוצה של השחקן והקבוצה 0- סיבוכיות זמן משוערכת של $O(\log^*(k))$ יחד עם איחוד הקבוצות (בשאר הפונקציות).
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k))$ משוערך, בממוצע על הקלט.

StatusType removePlayer(void *DS, int PlayerID)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- נחפש את השחקן בטבלת הערבול- במידה ולא נמצא נחזיר שגיאה.
- במידה ונמצא, נשמור את ערכי הקבוצה התוצאה והרמה שלו.
- הפעולות הבאות יתבצעו גם לקבוצת השחקן וגם לקבוצת כל השחקנים (הקבוצה 0).
- נמצא את הקבוצה, ניגש למערך התוצאות בתא המיוצג על ידי תוצאת השחקן.
- במידה ורמת השחקן היא 0, נעדכן את מונה שחקני הרמה 0, במידה ולא, נסיר את השחקן מעץ השחקנים לפי רמה.
- נעשה אותו דבר בדיוק גם בתא ה-0 במערך, המייצג את כל השחקנים בקבוצה ללא חשיבות לתוצאה.
- נסיר את השחקן מטבלת הערבול.

סיבוכיות זמן:

- חיפוש השחקן בטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- חיפוש הקבוצה של השחקן והקבוצה 0- סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- הסרת השחקן מהקבוצות- במקרה ה"גרוע" בו רמתו שונה מאפס- סיבוכיות זמן של $O(\log(n))$.
- הסרת השחקן מטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם שאר הפונקציות, בממוצע על הקלט.



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 5 מתוך 8

StatusType increasePlayerIDLevel(void *DS, int PlayerID, int LevelIncrease)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- נחפש את השחקן בטבלת הערבול- במידה ולא נמצא נחזיר שגיאה.
- במידה ונמצא, נשמור את ערכי הקבוצה התוצאה והרמה שלו.
- הפעולות הבאות יתבצעו גם לקבוצת השחקן וגם לקבוצת כל השחקנים (הקבוצה 0).
- נמצא את הקבוצה, ניגש למערך התוצאות בתא המיוצג על ידי תוצאת השחקן.
- במידה ורמת השחקן היא 0, נעדכן את מונה השחקני רמה 0, במידה ולא, נסיר את השחקן מעץ השחקנים לפי רמה.
- נעשה אותו דבר בדיוק גם בתא ה-0 במערך, המייצג את כל השחקנים בקבוצה ללא חשיבות לתוצאה.
- נעדכן את ערך רמת השחקן לרמה החדשה.
- נוסיף את השחקן מחדש לעץ הנמצא בתא של תוצאת השחקן, נוסיף אותו גם לעץ של כל השחקנים בקבוצה ללא חשיבות לתוצאה (התא 0 במערך).

סיבוכיות זמן:

- חיפוש השחקן בטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- חיפוש הקבוצה של השחקן והקבוצה 0- סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- הסרת השחקן מהקבוצות- במקרה ה"גרוע" בו רמתו שונה מאפס- סיבוכיות זמן של $O(\log(n))$
- הוספת השחקן לקבוצות- רמת השחקן כעת בהכרח שונה מאפס ולכן- סיבוכיות זמן של $O(\log(n))$
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם כל הפונקציות, בממוצע על הקלט.

בונוס (5 נק'):

תארו בחלק היבש כיצד ניתן היה לממש את הפעולה הנ"ל בסיבוכיות של $O(\log^*(k) + \log(n))$ משוערך (שימו לב שכאן הסיבוכיות אינה בממוצע על הקלט).

פיתרון:

השינוי העיקרי הינו מימוש טבלת הערבול של השחקנים באמצעות עצי AVL מאוזנים הממוינים לפי playerID במקום שימוש בשיטת השרשראות.

לאחר השינוי, חיפוש השחקן בטבלת הערבול ייעשה ב $O(\log(n))$ במקרה הגרוע כך שבסך הכול סיבוכיות הפונקציה תהיה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם שאר הפונקציות. נבחין כי סיבוכיות שאר הפונקציות לא תיפגע מן השינוי, כיוון שבממוצע בעקבות פקטור העומס שנשאר קבוע, החיפוש של שחקן בטבלה, הכנסה והוצאה נשאר $O(1)$ בממוצע, כיוון שגובה העצים בתאים הינו $O(\log \alpha)$, כאשר α הוא פקטור העומס והוא קבוע.



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 6 מתוך 8

StatusType changePlayerIDScore(void *DS, int PlayerID, int NewScore)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- נחפש את השחקן בטבלת הערבול- במידה ולא נמצא נחזיר שגיאה.
- במידה ונמצא, נשמור את ערכי הקבוצה התוצאה והרמה שלו.
- הפעולות הבאות יתבצעו גם לקבוצת השחקן וגם לקבוצת כל השחקנים (הקבוצה 0).
- נמצא את הקבוצה, ניגש למערך התוצאות בתא המיוצג על ידי תוצאת השחקן.
- במידה ורמת השחקן היא 0, נעדכן את מונה שחקני הרמה 0, במידה ולא, נסיר את השחקן מעץ השחקנים לפי רמה.
- נעדכן את ערך תוצאת השחקן לתוצאה החדשה.
- ניגש לתא במערך התוצאות המייצג את התוצאה החדשה של השחקן, במידה ורמת השחקן היא 0 מונה שחקני הרמה 0, אחרת נוסיף את השחקן מחדש לעץ.

סיבוכיות זמן:

- חיפוש השחקן בטבלת הערבול- סיבוכיות זמן של $O(1)$ בממוצע על הקלט.
- חיפוש הקבוצה של השחקן - סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- הסרת השחקן מהקבוצות- במקרה ה"גרוע" בו רמתו שונה מאפס- סיבוכיות זמן של $O(\log n)$
- הוספת השחקן לקבוצות- רמת השחקן כעת בהכרח שונה מאפס ולכן- סיבוכיות זמן של $O(\log n)$
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם שאר הפונקציות, בממוצע על הקלט.

StatusType getPercentOfPlayersWithScoreInBounds (void *DS, int GroupID, int score, int lowerLevel, int higherLevel, double * players)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- כל הפעולות המתוארות כאן תקפות גם עבור הקבוצה 0- קבוצת כל השחקנים, שתואמת לקלט groupID=0.
- נמצא את הקבוצה וניגש לעץ השחקנים בתא במערך המיוצג על ידי התוצאה הספציפית.
- נוסיף לעץ שני צמתי שחקני דמה המחזיקים את חסם הרמה העליונה עם תעודת זהות מקסימלית וחסם הרמה התחתונה עם תעודת זהות מינימלית, כך נדאג שצמתיים אלו יהיו מקסימלי ברמה העליונה ומינימלי ברמה התחתונה.
- נשתמש באלגוריתם select עבור שני צמתיים אלה.
- נשמור במשתנה את ההפרש בין האינדקסים של שני הצמתיים האלה, נבחין כי מכיוון שצמתי השחקנים בעלי רמה 0 לא נמצאים בעץ, במידה וחסם הרמה התחתונה שלנו הוא 0, נוסיף את מספר השחקנים ברמה אפס למשתנה שלנו.
- אם ערך המשתנה קטן או שווה לאפס, נחזיר שגיאה כיוון שאין שחקנים בטווח הרמות הללו.
- נסיר את שני צמתי הדמה מהעץ.



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 7 מתוך 8

- באותו אופן, נעשה את כל הפעולות הנ"ל עבור התא ה-0 במערך score_structure של הקבוצה הנ"ל (המתאר את כל השחקנים ללא חשיבות לתוצאה).
- נחזיר את אחוז ערך המשתנה הראשון מערך המשתנה השני.

סיבוכיות זמן:

- חיפוש הקבוצות הנדרשות - סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- הוספת צמתי דמה לעצי התוצאה הספציפית- סיבוכיות זמן של $O(\log(n))$.
- ביצוע פעולת select- סיבוכיות זמן של $O(\log(n))$.
- הסרת צמתי הדמה מעצי התוצאה הספציפית- סיבוכיות זמן של $O(\log(n))$.
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם שאר הפונקציות, בממוצע על הקלט.

StatusType averageHighestPlayerLevelByGroup(void *DS, int GroupID, int m, double * avgLevel)

נבצע את הפעולות הבאות:

- נבדוק תקינות הקלט, במידה ולא תקין נחזיר שגיאה.
- כל הפעולות המתוארות כאן תקפות גם עבור הקבוצה ה-0 קבוצת כל השחקנים, שתואמת לקלט groupID=0.
- נמצא את הקבוצה, במידה ומספר השחקנים בקבוצה קטן מ-m נחזיר שגיאה.
- נבדוק האם מספר השחקנים בקבוצה שווה למספר השחקנים בעלי הרמה אפס, במידה וכן נחזיר בממוצע אפס ונסיים.
- ניגש לתא 0 במערך התוצאות, התא שמייצג את כל השחקנים בקבוצה ללא חשיבות לתוצאה.
- נחשב את n_prime- מספר השחקנים בקבוצה שהם לא ברמה 0.
- נגדיר משתנה חדש- index_to_select שהינו האיבר בעל האינדקס המ-n_prime בעץ.
- נשתמש באלגוריתם select עבור האיבר ה-index_to_select, כך שתוך כדי ביצוע האלגוריתם אנו סוכמים את כל הרמות של הצמתים הקטנים מהאיבר הנ"ל.
- נשתמש באלגוריתם select עבור האיבר המקסימלי בעץ- תוך כדי ביצוע האלגוריתם אנו סוכמים את כל הרמות של הצמתים הקטנים מהאיבר הנ"ל- בעצם נקבל את סכום הרמות בכל העץ.
- נחזיר את הרמה הממוצעת שהוא הפרש הסכומים חלקי m.

סיבוכיות זמן:

- חיפוש הקבוצות הנדרשות - סיבוכיות זמן משוערכת של $O(\log^*(k))$.
- ביצוע פעולת select- סיבוכיות זמן של $O(\log(n))$.
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(\log^*(k) + \log(n))$ משוערכת ביחד עם שאר הפונקציות, בממוצע על הקלט.



מבני נתונים 234218 חורף תשפ"ב

גיליון רטוב מספר 2 –נתיב מאור 319002911, בר בן אבי 318176138
עמוד 8 מתוך 8

void Quit(void **DS)

נבצע את הפעולות הבאות:

- נשחרר את כל השחקנים מטבלת הערבוד של השחקנים.
- נשחרר את שלושת המערכים הנמצאים ב-UF.

סיבוכיות זמן:

- שחרור טבלת הערבוד- סיבוכיות זמן של $O(n)$.
- שחרור מערכי $\text{size} \text{ parent}$ - סיבוכיות זמן של $O(k)$.
- שחרור מערך הקבוצות- בכל קבוצה נשחרר את מערך התוצאות שלה, בנוסף נשחרר את עצי השחקנים שלה, נסמן ב- n_i את מספר השחקנים בקבוצה ה- i . מתקיים כי $\sum_{i=1}^k (n_i + 1) = O(n + k)$ ולכן סיבוכיות זמן של $O(n+k)$.
- כל שאר הפעולות- סיבוכיות זמן של $O(1)$.
- בסך הכול סיבוכיות הפונקציה $O(n + k)$ במקרה הגרוע.

הסבר על סיבוכיות הזמן המשוערכת של כל הפעולות יחד-

בכל הפעולות אנו מבצעים מספר קבוע של פעולות חיפוש/איחוד קבוצות במבנה ה-UF. סך סיבוכיות הזמן של החלק הנ"ל הוא $O(\log^*(k)) \cdot t$ כאשר t הוא מספר הפעולות. לכן התוספת לסיבוכיות המשוערכת של כל פונקציה מפעולות אלו היא $O(\log^*(k))$. נוסף על כך, בפעולות AddPlayer ו-RemovePlayer שבהן גודל טבלת הערבוד הדינאמית עלול להשתנות, התוספת לסיבוכיות המשוערכת של הפונקציות מפעולות אלו היא $O(1)$ כפי שלמדנו.

סיבוכיות מקום -

בכל רגע נתון אנו מחזיקים במבנה הנתונים שלנו:

- טבלת ערבוד של שחקנים המכילה n שחקנים- גודל טבלת הערבוד הינו $O(n)$.
 - שלושה מערכים בגודל k בתוך מבנה ה-UnionFind- סיבוכיות מקום של $O(k)$.
 - k מערכים בגודל scale - סיבוכיות מקום של $O(k)$.
 - $k \cdot \text{scale}$ עצי AVL כאשר סך כל הצמתים בכל העצים הוא לכל היותר n - סיבוכיות מקום של $O(n)$.
 - נציין כי הפעולות שבהן התבצע חיפוש בעץ מתבצעות ב- $O(\log(h)) = O(\log(n))$ כיוון שגודל העצים הינו לכל היותר n .
- לכן סיבוכיות המקום הינה $O(n + k)$ במקרה הגרוע, כאשר n הוא מספר השחקנים ו- k הוא מספר הקבוצות.