



מגשים:

ניצן שנדור 311158968

sandor@campus.technion.ac.il

בן גמליאל 305172991

bengamliel@campus.technion.ac.il

תיאור מבני הנתונים:

מבנה הנתונים מתבסס על עצי AVL המחזיקים איברים שונים או ממויינים באופן שונה. קיימת מחלקה Oasis שמנהלת את מבנה הנתונים כולו. היא מחזיקה 3 עצים שיאפשרו לה לבצע את הפעולות בסיבוכיות הזמן והמקום הנדרשות: עץ המחזיק שמחקנים הממויינים לפי מזהה, עץ המחזיק את השחקנים הממויינים לפי coins, ועץ המכיל שבטים. כל שבט מכיל עץ השומר את השחקנים ששייכים אליו.

תיאור המחלקות (החל מהמחלקה הפנימית ביותר עד למעטפת של כל מבנה הנתונים):

עץ AVL: עץ חיפוש בינארי מאוזן גנרי. נסמן את מספר הצמתים בעץ ב-n, וגובה המקסימלי של העץ הינו $h=O(\log(n))$, העץ תומך בפעולות הבאות:

- חיפוש: על פי אלגוריתם חיפוש בינארי שנלמד בכיתה – סיבוכיות זמן $O(\log(n))$.
- הכנסה: הכנסה לעץ ואיזון העץ לאחר מכן בעזרת גלגולים (LR,RR,RL,LL) כפי שנלמד בכיתה – סיבוכיות זמן $O(\log(n))$.
- הוצאה: הוצאה מהעץ ואיזון העץ לאחר מכן בעזרת גלגולים (LR,RR,RL,LL) כפי שנלמד בכיתה – סיבוכיות זמן $O(\log(n))$.
- איחוד: לפי אלגוריתם איחוד עצים אשר שנלמד בכיתה – כשאר ידוע שבשני העצים ביחד יש לכל היותר n איברים (כמספר השחקנים במשחק) ולכן סיבוכיות הזמן היא $O(n)$.
- החזרת מצביעים לאיברי העץ ממויינים במערך לפי הסדר שנקבע בהשוואת האיברים הגנריים. באמצעות מעבר על כל איברים העץ בסיור inorder (הפוך כי רוצים ממויינים בסדר יורד) והכנסת כל איבר למערך שמתקבל – סיבוכיות זמן $O(n)$.
- הריסה: שחרור של כל איבר בעץ לפי סיור – סיבוכיות זמן שחרור $O(1)$ ועבור כל האיברים $O(n)$.

סימון מספר השחקנים: n, מספר השבטים: k.

שחקן: מחלקה אשר מכילה מידע על שחקן בעל השדות הבאים: מזהה(id), מטבעות(coins), אתגרים שהושלמו(challengeComplete). ישנן מחלקות העוטפות את המחלקה הנועדו לאפשר השוואה בין מצביעים של שחקנים על מנת שיתאפשר לנו לשמור עותק אחד לשחקן ומסייעת לנו בניהול הזיכרון:

- Player_score: מסייעת לנו לבצע את ההשוואה בין שחקנים נעשית בסדר הבא תחילה על פי כמות מטבעות ולאחר מכן לפי המזהה. כאשר המחלקה מחזיקה מצביע לשחקן עצמו ואינה אחראית לשיחרור.
- Player_ptr: יורשת מ-Player_score. המזהה השווה בין איברי המחלקה לפי מזהה השחקן בעלת מצביע לשחקן בהריסה האובייקט המחלקה משחררת גם את השחקן.

Clan: בעל השדות הבאים: מחלקה המכילה עץ AVL אשר מכיל את השחקנים הנמצאים בשבט (מסוג Player_score), שדה id אשר מייצג את מספר הקלאן, ושדה אשר מכיל את השחקן הטוב ביותר בקלאן מסוג (Player_score) המחלקה תומכת בפעולות הבאות (פירוט אופן העובדה על עץ AVL כמו שמופרט בפירוט על מחלקה AVL):

- הוספת שחקן: הוספה לעץ AVL – סיבוכיות זמן $O(\log(n))$

מגשים:

ניצן שנדור 311158968

sandor@campus.technion.ac.il

בן גמליאל 305172991

bengamliel@campus.technion.ac.il




- הוצאת שחקן: מחיקה מעץ AVL – סיבוכיות זמן $O(\log(n))$.
- בנוסף שומר את השחקן בעל התוצאה המקסימלית.
- קבלת השחקנים ממויינים במערך לפי סדר המיון של העץ (לפי coins) - סיבוכיות זמן $O(n)$.

Oasis: המחלקה מכילה 4 שדות, 3 עצי AVL ומצביע לטיפוס מסוג Player

- $AVL < Player_ptr >^*$ עץ המכיל את כל השחקנים במערכת (מסוג Player_ptr)
- $AVL < Clan >^*$ עץ אשר מכיל את כל השבטים במערכת.
- $AVL < Player_score >^*$ עץ אשר שומר את התוצאה של כל השחקנים במערכת ע"ס כמות ה-coins שלהם (ומיון משני ע"ס מזהה).
- $Player^* \text{ best}$ מצביע לטיפוס מסוג Player אשר שומר את השחקן בעל התוצאה המקסימלית בכל המערכת.

המחלקה תומכת בפעולות הבאות ומהווה מעטפת לכלל המחלקות במערכת:

אופן העובדה על עצי AVL כמו שמופרט בפירוט על מחלקה AVL

- יצירת שחקן והוספתו לעץ - סיבוכיות זמן $O(\log(n))$. 
- יצירת שבט והוספתו לעץ השבטים - סיבוכיות זמן $O(\log(n))$.
- $joinClan$: חיפוש השחקן בעץ השחקנים (סיבוכיות זמן $O(\log(n))$) אם הוא כבר בשבט כלשהו אז סיימנו. חיפוש השבט המתאים בעץ השבטים (סיבוכיות זמן $O(\log(k))$) במידה והשבט ריק שינוי השבט (סיבוכיות זמן $O(1)$), הכנסה לעץ (סיבוכיות זמן $O(\log(n))$) – סהכ סיבוכיות זמן $O(\log(k) + \log(n))$. 
- $completeChallange$: חיפוש השחקן בעץ השחקנים. הוצאתו מעץ השחקנים הממויין לפי coins (סיבוכיות $O(\log(n))$), אם קיים בשבט, הוצאתו מהשבט (גישה ישירה מהשחקן אל השבט) – מציאת השבט בסיבוכיות $O(1)$ והסרתו מהשבט בסיבוכיות $O(\log(n))$. עדכון השדות המתאימים (סיבוכיות זמן $O(1)$) והכנסתו מחדש לעצים – הכנסה לעץ השחקנים הממויין לפי coins (סיבוכיות זמן $O(\log(n))$) ובמידה והיה בשבט הכנסו מחדש לעץ של השבט (סיבוכיות זמן $O(\log(n))$). סהכ סיבוכיות זמן $O(\log(n))$. 

- $getBestPlayer$: מתחלק ל-2 במערכת כולה או בקלאן ספציפי
 - אם בכל המערכת: החזרת האיבר המקסימלי של Oasis – סיבוכיות זמן $O(1)$ (שמור בשדות של Oasis).
 - אם בשבט מסויים: חיפוש השבט עצמו (סיבוכיות זמן $O(\log(k))$) וחזרת השחקן הטוב ביותר (סיבוכיות זמן $O(1)$ - שמור בשדה של השבט) סהכ סיבוכיות זמן $O(\log(k))$.
- $getScoreboard$: מתחלק לשני מקרים, כל המערכת או בשבט בודד

- במקרה של כל המערכת בסיבוכיות של $O(n)$ כמספר השחקנים במערכת וסידורם במערך. סידור של המזהים של השחקנים במערך אותו יש להחזיר $O(n)$. סיבוכיות מקום נוסף – $O(n)$.

מגשים:

ניצן שנדור 311158968

sandor@campus.technion.ac.il

בן גמליאל 305172991

bengamliel@campus.technion.ac.il

- במקרה של שבט בודד חיפוש השבט המתאים $O(\log(k))$ ומספר השחקנים בשבט הינו מספר קטן שווה למספר השחקנים הכולל במערכת נקרא לו $n \geq c$ וסידורם במערך הינו $O(c)$ סה"כ קיבלנו $O(\log(k)+c)$. סיבוכיות מקום נוסף – $O(c)$.

uniteClans: חיפוש שני השבטים אותם יש לאחד לשבט אחד בעץ השבטים (סיבוכיות זמן $O(\log(k))$). איחוד שני עצים שלשניהם ביחד לכל היותר n איברים (סיבוכיות זמן $O(n)$). הוצאה של שני השבטים הישנים והכנסת השבט החדש – כל אחת מהפעולות בסיבוכיות זמן $O(\log(k))$. סך הכל סיבוכיות זמן כוללת של $O(n + \log(k))$.



Quit: הריסת כל מבנה הנתונים – הריסת שני עצי השחקנים (סיבוכיות זמן הריסת עץ $O(n)$)-מעבר על כל השחקנים בעץ.

הריסת עץ השבטים: הריסת כל איבר בעץ כוללת הריסת עץ של שחקנים. מתקיים

$\sum_{i=1}^k \text{size}(\text{Clan}_i) = n$ לכן סיבוכיות הזמן להריסת עצים אלו $O(n)$. הריסת עץ השבטים עצמו $O(k)$ לכן סיבוכיות הזמן הכוללת $O(n+k)$.

סיבוכיות מקום:

כל שחקן מכיל מספר קבוע של שדות – $O(1)$

כל עץ מכיל מספר קבוע של שדות בכל node ואת האיבר הנשמר בו. ולכן סיבוכיות המקום חסומה ע"י מספר האיברים בעץ. החזקת שני עצים המכילים את כל השחקנים – $O(n)$

כל השדות בשבט מלבד העץ לוקחים מקום קבוע – $O(1)$, העץ מכיל שחקנים, כאשר $\sum_{i=1}^k \text{size}(\text{Clan}_i) = n$ לכן כל השבטים ביחד יקחו $O(k+n)$

עץ השבטים – מכיל את כל השבטים – $O(k+n)$.

בסך הכל סיבוכיות מקום של מבנה הנתונים: $O(n+k)$.