

תיאור מערכת

המערכת נקראת Oasis. המערכת ממומשת כמחלקה אשר מכילה את השדות הבאים:

- המחלקה שומרת מצביע לשחקן "הכי טוב". זהו השחקן שהשלים הכי הרבה אתגרים עד כה מבין כל שחקני המערכת. שדה זה שימושי במקרה של הפעלת הפונקציה `getBestPlayer` עם ארגומנט `clanID<0`.
- במערכת שומרים עץ חיפוש מאוזן של שחקנים. המידע בכל צומת בעץ מכיל מצביע לשחקן אחד במערכת. השחקנים בעץ מסודרים על בסיס מספרי הזיהוי שלהם.
- במערכת נשמר גם עץ חיפוש מאוזן של מצביעים לשחקנים לפי מטבעות. בעץ זה המידע בכל צומת מכיל מצביע לשחקן אחד במערכת. השחקנים בעץ זה מסודרים על ידי מספר המטבעות שצברו עד כה כמיון ראשי, ועל פי מספרי הזיהוי שלהם כמיון משני. המיון המשני הכרחי מאחר ואין מניעה שלשני שחקנים שונים יהיה מספר מטבעות שווה, אך בהכרח לשני שחקנים שונים יהיה מספר זיהוי שונה. עץ זה יהיה שימושי במקרה של הפעלת הפונקציה `getScoreBoard` עם ארגומנט `clanID<0`.
- במערכת שומרים עץ חיפוש מאוזן של שבטים. המידע בכל צומת בעץ מכיל מצביע לשבט אחד במערכת. השבטים מסודרים על בסיס מספרי הזיהוי שלהם.

לכל שחקן שומרים את מספר הזיהוי שלו המשמש למציאת השחקן הנדרש בהפעלת מגוון פונקציות המערכת. בנוסף, שומרים לכל שחקן את מספר המטבעות שהשיג עד כה, את מספר האתגרים שהשלים, ומצביע לשבט שלו. סך הכול סיבוכיות המקום שלוקח כל שחקן היא $O(1)$.

לכל שבט שומרים את מספר הזיהוי שלו המשמש למציאת השבט הנדרש בהפעלת מגוון פונקציות המערכת. בכל שבט נשמר מצביע לשחקן "הכי טוב בשבט", כלומר השחקן שמשויך לשבט שהשלים הכי הרבה אתגרים. בנוסף, שומרים לכל שבט שני עצי חיפוש מאוזנים:

- עץ חיפוש מאוזן של שחקנים, המכיל מצביעים לשחקנים המשויכים לשבט כאשר המיון הוא לפי מספרי זיהוי השחקנים. סיבוכיות המקום של העץ במקרה הגרוע $O(n)$.
- עץ חיפוש מאוזן של שחקנים לפי מטבעות, המכיל מצביעים לשחקנים המשויכים לשבט כך שהמיון הראשי הוא לפי מטבעות והמשני לפי מספר זיהוי. סיבוכיות המקום של עץ זה במקרה הגרוע הוא גם $O(n)$.

עצים אלו דומים במשמעותם לעצים של המערכת עצמה, ההבדל הוא שעבור כל שבט נשמרים השחקנים משויכים לשבט בלבד, ולא כל השחקנים במערכת.

במקרה הגרוע, שבו כל השחקנים במערכת משויכים לאותו השבט, סך כל הסיבוכיות של שבט הוא $O(n)$.

ניתוח סיבוכיות מקום המערכת:

מצביע לשחקן הכי טוב במערכת – $O(1)$.

עץ חיפוש מאוזן של שחקנים – סיבוכיות מקום של כל שחקן $O(1)$. כל שחקן נשמר בצומת אחד בעץ. לכן בהינתן n שחקנים סך סיבוכיות המקום של העץ היא $O(n)$.

עץ חיפוש מאוזן של שבטים – כל שבט נשמר בצומת אחד בעץ. בהינתן k שבטים אזי בעץ $O(k)$ צמתים. בכל השבטים הללו ביחד נשמרים מצביעים לכל היותר לכל השחקנים במערכת. מאחר ושחקן יכול להיות משויך לשבט אחד בלבד בזמן נתון, אזי שחקן יופיע פעמיים בלבד בעץ השבטים – פעם אחת בעץ השחקנים של השבט שלו, ופעם נוספת בעץ השחקנים לפי מטבעות של השבט שלו. הסיבוכיות לכך היא $O(n)$. לכן סך כל סיבוכיות המקום שלוקח עץ השבטים הוא $O(n+k)$.

עץ חיפוש מאוזן של שחקנים לפי מטבעות – מכיל כמות מידע זהה לעץ שחקנים, ההבדל היחידי בין שני העצים היא פרמטרי המיון השונים של השחקנים. על כן מאותן סיבוכיות המקום היא $O(n)$.

לסיכום, סך סיבוכיות המקום במערכת היא חיבור של כל הנתונים הנ"ל, כלומר $O(1+n+n+k+n) = O(n+k)$.

פונקציית init:

הפונקציה אינה מקבלת פרמטרים, יוצרת את אובייקט המערכת ומחזירה מצביע אליו. בעת הקצאת הזיכרון למערכת הפונקציה בודקת האם הייתה בעיה בהקצאה, אם כן היא מטפלת בכך בהתאם.

הפונקציה קוראת לבנאי של המחלקה oasis שמכילה 4 שדות, כולם מצביעים לטיפוסים שונים – מצביע לעץ שחקנים, מצביע לעץ שבטים, מצביע לעץ שחקנים מסודרים לפי מטבעות ומצביע לשחקן הכי טוב במערכת. מאחר ובעת התיחול של המערכת עדיין אין נתונים, הבנאי מתחיל את כל שלושת המצביעים לעצים שיציעו לעץ ריק, ואת המצביע לשחקן הכי טוב ל-NULL. הסיבוכיות לתיחול עץ ריק או מצביע ל-NULL היא $O(1)$, ולכן סיבוכיות הזמן של הפונקציה היא גם כן $O(1)$.

פונקציית quit:

הפונקציה מקבלת מצביע לאובייקט המערכת. במידה והמצביע אינו מצביע ל-NULL, הפונקציה קוראת להורס המערכת. הורס המערכת עצמו קורא 3 פעמים לפונקציית הריסה של עץ, פעם אחת עבור כל עץ ששמור במערכת.

הריסת עץ מסוים מחייבת מעבר על כל צומת בעץ והריסתו. עבור עץ השבטים, מכיוון שעלינו לעבור על כל שבט, ובתוכו לשחרר את שני עצי השחקנים של השבט. סך הכול קיימים k שבטים שביחד מכילים עד n שחקנים. עבור עצי השחקנים, שחרור כל צומת בעץ מתבצע ב- $O(1)$, ולכן השחרור עבור כל השחקנים הוא $O(n)$. לכך נוסיף שקיימים k עצי שבטים המכילים ביחד את כל השחקנים, ולכן סיבוכיות הזמן של הפונקציה $O(n+k)$.

פונקציית addPlayer:

הפונקציה מקבלת את אובייקט המערכת, מספר זיהוי של שחקן ומספר מטבעות התחלתי. הפונקציה מבצעת חיפוש במערכת אם השחקן קיים. במידה ולא קיים אזי פונקציית החיפוש מחזירה את הצומת האחרון במסלול החיפוש. לצומת זו הפונקציה מוסיפה כבן את השחקן החדש בשני עצי השחקנים, ולאחר מכן מאזנים את העצים. בנוסף, מתבצעת בדיקה האם השחקן החדש הוא הכי טוב במערכת. אם התקבלו קלטים לא תקינים, אם השחקן כבר נמצא במערכת, או אם הייתה שגיאת הקצאת זיכרון המערכת מחזירה הודעת שגיאה מתאימה.

בניית אובייקט שחקן חדש מתבצע ב- $O(1)$. חיפוש השחקן בכל אחד משני עצי השחקנים של המערכת מתבצע ב- $O(\log n)$ בזכות העובדה שהעצים הם עצי חיפוש מאוזנים. הכנסת השחקן לכל עץ לוקחת גם $O(1)$ זמן, ולאחר מכן מבצעים גלגול בעת הצורך הלוקח גם $O(1)$. בדיקה האם השחקן הוא הכי טוב במערכת ועדכון המערכת בהתאם לוקח גם $O(1)$ בזכות העובדה שקיים מצביע במערכת לשחקן הכי טוב המאפשר השוואה מידית ללא צורך בחיפוש השחקן הטוב ביותר. לכן סיבוכיות הזמן של הפונקציה היא $O(\log n)$.

פונקציית getBestPlayer:

הפונקציה מקבלת את אובייקט המערכת, מספר זיהוי של שבט ומצביע למספר זיהוי של שחקן. הפונקציה מפנה את המצביע של מספר הזיהוי המתקבל לשחקן הטוב ביותר בשבט שהתקבל. אם מספר הזיהוי של השבט שווה 0 אזי הפונקציה תחזיר את השחקן הטוב ביותר בכל המערכת, ואם אינו חוקי, או במידה ואין שחקנים בשבט אזי הפונקציה תחזיר -1.

חיפוש השבט המתאים לוקח $O(\log k)$ בזכות העובדה שעץ השבטים הוא עץ חיפוש מאוזן. בדיקת מספר השבט הנקלט ובמידת הצורך לקיחת השחקן הכי טוב בכל המערכת לוקחים $O(1)$, ולכן סיבוכיות הזמן של הפונקציה היא $O(\log k)$.