

# מבני נתונים

תרגיל רטוב 1 – חלק יבש

מגישים: סיון זכריה, ת.ז. 319124269

גל ישי, ת.ז. 209926252

תיאור מבנה הנתונים – מבנה הנתונים יכלול עץ מדינות שממויין לפי ID (של מדינה), עץ קבוצות שממויין לפי ID (של קבוצה) ועץ מתחרים שממויין לפי ID של מתחרים. בנוסף, לכל קבוצה יהיו 6 עצים כך ש-3 עצים מתוכם יהיו ממויינים לפי ID של השחקנים בקבוצה ויהיו מחולקים לשלוש ושלושת העצים האחרים יהיו עצים הממויינים לפי כוחות השחקנים של כל שלשה (כלומר לכל עץ שממויין לפי ID ומתאר שלשה של מתחרים יהיה עץ שממויין לפי כוחות ומתאים למתחרים של אותה שלשה). בנוסף, לכל קבוצה יהיה עץ שממויין לפי כוחות כל השחקנים שבקבוצה. יהיה מצביע דו כיווני בין מתחרה שמופיע באחד משלושת העצים הממויינים לפי ID לבין החוליה המתאימה למתחרה זה בשלשה המתאים בעצי הכוחות, יהיה מצביע דו כיווני בין מתחרה שמופיע באחד משלושת העצים הממויינים לפי ID לבין חוליה המתארת את המתחרה בעץ הכוחות של הקבוצה.

פירוט מבנה הנתונים -

-  $m\_Countries$  – עץ AVL שצמתיו ייצגו מדינות ויהיה ממויין לפי ID של מדינה.

-  $m\_Teams$  – עץ AVL שצמתיו ייצגו קבוצות ויהיה ממויין לפי ID של קבוצה.

-  $m\_Players$  – עץ AVL שצמתיו ייצגו מתחרים ויהיה ממויין לפי ID של מתחרה.

נשים לב שכל מדינה בפרט תכיל ID של מדינה, מספר מדליות של המדינה, מספר מתחרים שיש בה ועץ עם הקבוצות שיש בה.

נשים לב שכל קבוצה בפרט תכיל – ID של קבוצה, סוג ספורט, מצביע למדינה אליה היא מצטרפת, 3 עצי AVL של המתחרים שנמצאים בה שממויינים לפי ID ומחולקים לשלוש ו-3 עצי AVL נוספים כך שכל אחד מהם יכיל את השחקנים בכל שלשה שמתאימה לו אך יהיה ממויין לפי כוחות המתחרים בשלשה המתאימה לו (נשים לב שיתכנו כמה מתחרים עם גודל כוח זהה לכן נבצע מיון ראשוני לפי כוח ומיון משני לפי ID בעץ זה). בנוסף, בכל שלשה כזאת נשמור מצביעים ל-3 ערכי  $max\ min$  בשלוש לפי ID ובשלוש שממויינות לפי הכוחות של המתחרים בכל שלשה מתאימה וניעזר בערכים אלו בפונקציה  $austerity\_measures$ . (העדכונים עבור השלוש יעשו רק בעץ הקבוצות של המדינה ולא בעץ הקבוצות הכללי).

נשים לב שכל מתחרה בפרט יכיל ID של מתחרה, גודל הכוח שלו מספר הקבוצות שהוא מתחרה בהן, סוג ספורט, מצביע למדינה בה הוא מתחרה, ומצביעים לקבוצות בהן הוא מתחרה (כאשר נתון שיש לכל היותר 3 כאלו).

בנוסף, יהיה מצביע דו כיווני בין שחקן בכל שלשה של ID בעץ הקבוצה שלו לבין החוליה שמייצגת את אותו שחקן בשלשה שממויינת לפי כוחות ומתאימה לאותה שלשה של מתחרים שממויינת לפי ID, יהיה מצביע דו כיווני בין שחקן בשלשה שממויינת לפי ID לבין אותו שחקן בעץ הכוחות הכללי של הקבוצה ויהיה מצביע בין השחקן בשלשה של ID לבין חוליה שמתאימה לו בעץ הכללי של המתחרים שממויין לפי ID.

מימוש הפעולות והוכחת דרישות סיבוכיות הזמן –

1.  $olympics\_t()$  – נאתחל את העצים להיות ריקים (בהתחלה אין במערכת מדינות, נבחרות ומתחרים). פעולה זו מתבצעת ב- $O(1)$  ולכן סך הכל סיבוכיות הזמן בפעולה היא  $O(1)$ .

2.  $virtual \sim olympics\_t()$  – ישנם 3 סוגי עצים במערכת – עץ מדינות שמכיל לכל היותר  $k$  מדינות, עץ קבוצות כללי שמכיל לכל היותר  $m$  קבוצות ועץ שחקנים שמכיל לכל היותר  $n$  שחקנים. בכל קבוצה במדינה מסויימת יש 3 עצים שממויינים לפי ID של המתחרים בקבוצה ו-3 עצים שממויינים לפי כוחות המתחרים כאשר השחקנים בכל שלשה כזאת נקבעים לפי השלוש של השחקנים בעצים הממויינים לפי ID ויש עץ כוחות של שחקנים בכל קבוצה. נשים לב שסך הכל יש  $O(n)$  שחקנים במערכת מכל הקבוצות (ובכל השלוש) לכן סך הכל סיבוכיות הזמן בפעולה זו היא  $O(n + k + m)$ .

3. *add\_country* – תחילה נחפש איבר עם *countryID* בעץ המדינות הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log k)$ . אם קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נכניס את המדינה עם ID זה לעץ המדינות, פעולה זו מתבצעת ב- $O(\log k)$ . נתון שבעת הכנסת מדינה אין נבחרות ומתחרים ששייכים לה לכן אין צורך לבצע הכנסות לשאר העצים. סך הכל סיבוכיות הזמן בפעולה זו היא  $O(\log k)$ .

4. *remove\_country* – תחילה נחפש איבר עם *countryID* בעץ המדינות הממויין לפי ID. חיפוש זה מתבצע ב- $O(\log k)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נבדוק שאין בה מתחרים או נבחרות בתוספת סיבוכיות  $O(1)$  ע"י בדיקת השדה של מספר המתחרים בה ובדיקה האם עץ הקבוצות שלה ריק באמצעות מצביע בין מדינה לעץ הקבוצות שלה. אם יש בה מתחרים או נבחרות נחזיר *FAILURE*. אחרת, נסיר את המדינה מעץ ה-AVL ב- $O(\log k)$  לפי ההרצאה. סך הכל סיבוכיות הזמן בפעולה זו היא  $O(\log k)$ .

5. *add\_team* – תחילה נחפש איבר עם *countryID* בעץ המדינות הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log k)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נחפש איבר עם *teamID* בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log m)$ . אם קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נכניס קבוצה זו לעץ הקבוצות הכללי ולעץ הקבוצות של המדינה ב- $O(\log m)$  (כי כבר מצאנו את המדינה) ונעדכן מצביע מהקבוצה להצביע על המדינה שאליה היא שייכת בתוספת סיבוכיות  $O(1)$ . סך הכל סיבוכיות הזמן בפעולה זו היא  $O(\log m + \log k)$ .

6. *remove\_team* – תחילה נחפש איבר עם *teamID* בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log m)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נמצא את הקבוצה המתאימה שמוצבעת ע"י המדינה אליה היא שייכת ב- $O(1)$  באמצעות מצביע מקבוצה למדינה המתאימה לה (שמתחזקים ב-*add\_team*) ומהמדינה לעץ קבוצות שלה. נבדוק שאין בה מתחרים ע"י בדיקה אם השלשה הראשונה של המתחרים (מבין אלו הממויינות לפי ID) ריקה (אם יש מתחרים יהיה לפחות מתחרה 1 בשלשה הראשונה כי השלוש ממויינות לפי ID). בדיקה זו מתבצעת ב- $O(1)$  באמצעות מצביעים. אם יש מתחרים בקבוצה נחזיר *FAILURE*. אחרת, נסיר את הקבוצה מעץ הקבוצות הכללי ב- $O(\log m)$  ומעץ הקבוצות של המדינה הספיציפית ב- $O(\log m)$  (כאשר ניגש אליו בתוספת סיבוכיות  $O(1)$  כפי שתיארנו קודם). סך הכל הסיבוכיות בפעולה זו היא  $O(\log m)$ .

7. *add\_contestant* – תחילה נחפש איבר עם *countryID* בעץ המדינות הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log k)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נחפש איבר עם *contentantID* בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log n)$ . אם קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נוסיף את המתחרה לעץ המתחרים הכללי ב- $O(\log n)$  ונגדיל את מספר המתחרים במדינה בתוספת סיבוכיות של  $O(1)$  לאחר שמצאנו את המדינה. (נשים לב שלא מעדכנים את עצי השלוש, למשל לא מעדכנים את עצי השלוש שממויינים לפי ID, כי הם שייכים לקבוצה מסויימת וכרגע הוספנו מתחרה למערכת ולא לקבוצה מסויימת). סך הכל סיבוכיות הזמן בפעולה זו היא  $O(\log n + \log k)$ .

8. *remove\_contestant* – תחילה נחפש איבר עם *contentantID* בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log n)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, נבדוק שהוא לא פעיל בנבחרת ב- $O(1)$  באמצעות בדיקה האם המצביעים של השחקן לקבוצות שהוא פעיל בהן הוא *nullptr* (כאשר נתון ששחקן יכול להשתתף לכל היותר ב-3 קבוצות לכן בדיקות אלו מתבצעות ב- $O(1)$ ). אם השחקן פעיל בנבחרת נחזיר *FAILURE*. אחרת, נוציא אותו מעץ השחקנים הכללי ב- $O(\log n)$ . נקטין את מספר המתחרים במדינה ב- $O(1)$ . (אין צורך להוציא אותו מעצי השלוש כי הוא לא פעיל בנבחרת ולכן לא יופיע בהן). סך הכל פעולה זו מתבצעת ב- $O(\log n)$ .

9. `add_contestant_to_team` – תחילה נחפש איבר עם `contentantID` בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log n)$  ונחפש איבר עם `teamID` בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log m)$ . אם אחד מהם לא קיים נחזיר `FAILURE`. אחרת, ניגש למדינה אליה הקבוצה שייכת בתוספת סיבוכיות  $O(1)$  באמצעות מצביע ששמור לה למדינה אליה היא שייכת. אם המדינה שמצאנו והמדינה שהשחקן שייך אליה שונים או אם סוג הספורט של השחקן ושל הקבוצה שונים או אם המתחרה משתתף כבר ב-3 נבחרות נחזיר `FAILURE` (בדיקות אלו מתבצעות ב- $O(1)$  באמצעות שדות של קבוצה ושל שחקן).

אחרת, נחפש את הקבוצה מבין הקבוצות של המדינה ב- $O(\log m)$  (כאשר כבר מצאנו את המדינה לפני זה) ונבדוק אם הוא נמצא באחת מעצי השלשות של ה-ID של הקבוצה ב- $O(\log n)$ , אם כן נחזיר `FAILURE`. אחרת, נכניס אותו לעץ הכוחות של הקבוצה ב- $O(\log n)$ .

בנוסף, נכניס אותו לשלשה המתאימה מעצי השלשות ב-ID, ואז לשלשה המתאימה לה בעצי השלשות של הכוחות. (\*) נשים לב שבהכנסה של השחקן לשלשה המתאימה נשמור על כך (שגדלי השלשות יהיו שווים) או (שגודל השלשה האמצעית שווה לגודל השלשה הימנית וגם השלשה השמאלית (עם ה-ID הכי קטנים) גדולה ב-1 מגודל זה) או (שגודל השלשה השמאלית שווה לגודל השלשה האמצעית וגודל זה גדול ב-1 מגודל השלשה הימנית). נבדוק לאיזה שלשה השחקן אמור להתכנס כך שישמר המיון לפי ה-ID ע"י בדיקת ערכי `mini max` של ה-ID בעצי השלשות של ה-ID, כאשר בדיקה זו מתבצעת ב- $O(1)$  כי אנחנו שומרים ערכים אלו ומעדכנים אותם כשיש שינויים בשלשות של קבוצה. אם הכנסת השחקן לשלשה מפרה את (\*) אז נבצע העברות בין השלשות כך ש(\*) ישמר.

למשל אם בעצי השלשות מופיעים ה-ID הבאים – 234,56,78 ונרצה להכניס שחקן עם  $ID = 1$  כאשר ID זה קטן מהערך המינימלי בשלשה השמאלית לכן השחקן אמור להיכנס אליה. במקרה זה נכניס אותו לשלשה השמאלית ב- $O(\log n)$  ונקבל 1234,56,78 שלא שומר על (\*) לכן נעביר את 4 (המקסימלי בשלשה השמאלית) לשלשה האמצעית ואת 6 (המקסימלי בשלשה האמצעית) לשלשה הימנית (כאשר מציאת ערכי `max` או `min` בשלשות מתבצעים ב- $O(1)$  כי אנחנו שומרים אותם ומעדכנים אותם בכל שינוי של השלשות). כל הוצאה משלשה והכנסה לשלשה אחרת מתבצעת ב- $O(\log n)$ .

באופן דומה בכל הכנסת שחקן נבצע את ההעברות הדרושות שיתבצעו ב- $\log n$  באופן דומה (כאשר יש מספר סופי של מקרים שדורשים העברות ושל העברות).

עבור מתחרים שהועברו נבצע עדכונים גם בעצי השלשות שממויינים לפי כוחות כאשר השלשה שאליהם הם יועברו נקבעת לפי באיזה שלשה נמצא המתחרה בעצי השלשות הממויינים לפי ID ונוכל לגשת לצמתים המתאימים ב- $O(1)$  באמצעות מצביעים ממתחרים בעצי שלשות ה-ID לבין המתחרים המתאימים בעצי שלשות הכוחות וגם במקרה זה המעברים יתבצעו ב- $O(\log n)$ .

נעדכן לשחקן שנוסף ולשחקנים שהועברו בין עצי השלשות (אם יש כאלו) לאיזה קבוצה הם נכנסו ב- $O(1)$  לאחר שמצאנו לאיזה קבוצה הם נכנסים.

בנוסף, נעדכן את המצביע בין המתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ של השלשה שממויינת לפי כוחות ולהפך (מצביע דו כיווני). נעדכן את המצביע ממתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ הכוחות של הקבוצה ולהפך (מצביע דו כיווני) בתוספת סיבוכיות של  $O(1)$  לאחר שמצאנו את הצמתים המתאימים.

נקרא לפונקציית עזר `updateMinMax` שמעדכנת עד שלושה ערכי `maxi min` בכל שלשה (נשתמש בעדכונים אלו בפונקציה האחרונה כדי לבצע אותה בסיבוכיות הנדרשת). כל עדכון של עד 3 ערכי `min/max` מתבצע ב- $O(\log n)$  באמצעות מציאת ערך זה ב- $O(\log n)$  (כשהמינימום הוא האיבר הכי שמאלי בעץ והמקסימום הוא האיבר הכי ימני בעץ) ואז נבצע הסרה של הצומת ב- $O(\log n)$  מכל שלשה ונחשב את ה-`min/max` כעת (אם נשארו איברים בשלשה) ועבור ערכי ה-`min/max` הבאים נוציא את הצומת שהיה ה-`min/max` ב- $O(\log n)$  ונחשב את ערך ה-`min/max` (אם נשארו איברים בשלשה). פעולה זו מתבצעת מספר סופי של פעמים ויש 6 שלשות (שממויינות לפי ID או לפי כוח של מתחרים בכל שלשה של ID). לכן סיבוכיות פעולות אלו היא  $O(\log n)$ . נכניס את האיברים שהוצאנו, כשכל הכנסה כזו מתבצעת ב- $O(\log n)$  ויש מספר סופי של הכנסות של איברים אלו כי מספר ההוצאות סופי כפי שתיארנו. לכן, סיבוכיות פעולות אלו היא  $O(\log n)$ . בהכנסות האיברים נעדכן את המצביעים הדו כיוונים בין המתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ של השלשה שממויינת לפי כוחות ולהפך (מצביע דו כיווני) בתוספת סיבוכיות של  $O(1)$ .

נגדיל את מספר השחקנים בקבוצה ב $O(1)$ . נשמור במצביע הצבעה מהשחקן לקבוצה ב $O(1)$ .  
סך הכל פעולה זו מתבצעת ב $O(\log n + \log m)$ .

10. `remove_contestant_from_team` — תחילה נחפש איבר עם `contentantID` בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב $O(\log n)$ .  
נחפש איבר עם `teamID` בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב $O(\log m)$ . אם אחד מהם לא קיים נחזיר `FAILURE`. אחרת, נבדוק אם לאחת מהקבוצות שאליה שייך השחקן יש אותו ID כמו לקבוצה שהתקבלה כפרמטר, אם לא – נחזיר `FAILURE` (בדיקה זו מתבצעת ב $O(1)$  כי נתון ששחקן יכול להשתתף בעד 3 נבחרות ואנחנו שומרים מצביעים לקבוצות בהן הוא משתתף לכן נבדוק ID של כל אחת מהן וזה מספר סופי של קבוצות לכן זה מתבצע ב $O(1)$ ).  
אחרת, נמצא את השחקן בעץ הכוחות של הקבוצה ב $O(\log n)$  (כאשר מצאנו לפני זה את הקבוצה ב $O(\log m)$ ). באמצעות מצביע מצומת זה לצומת המתאים למתחרה זה בשלשה של ID נמצא אותו שם ב $O(1)$  ונשמור באיזה שלשה הוא ובאמצעות מצביע מצומת זה לצומת המתאים לו בשלשה המתאימה של הכוחות נוכל למצוא את הצומת המתאים לו שם. נסיר אותו מעץ הכוחות של הקבוצה ב $O(\log n)$ . נסיר אותו מהשלשה שלו שממוינת לפי ID ומהשלשה שלו שממוינת לפי כוחות כאשר נשים לב שנרצה לשמור על התנאי המתואר ב $(*)$  בפונקציה 9. לכן, אם התנאי יופר נבצע מעברים בין שלשות. (כאשר יש מספר סופי של מקרים של הפרת התנאי בהסרת שחקן אם התנאי התקיים לפני זה ומספר סופי של מעברים שצריך לעשות כדי לתקן אותם כך שכל הסרה משלשה והוספה לשלשה אחרת מתבצע ב $O(\log n)$  כפי שמתואר בפונקציה 9 ואפשר לדעת את מי להעביר ב $O(1)$  באמצעות ערכי `maxi min` ששומרים עבור השלשות ומעדכנים כשיש שינוי בשלשות).  
למשל אם השלשות של ID הן 12,34,56 ומסירים את 4 מהקבוצה ב $O(\log n)$  אז יתקבל המצב 12,3,56 שמפר את התנאי ב $(*)$ . במצב זה נעביר את 5 (הID המינימלי מהשלשה הימנית) לשלשה האמצעית (כאשר הסרה משלשה מתבצעת ב $O(\log n)$  והכנה של שלשה מתבצעת ב $O(\log n)$  ונקבל את השלשות 12,35,6 ששומרות על התנאי ב $(*)$ ).  
באופן דומה נפעל במצבים אחרים שמפרים את התנאי ב $(*)$ .  
המעברים בשלשות של הכוחות נקבעים לפי המעברים בשלשות של ID וכל מעבר מתבצע ב $O(\log n)$  באופן דומה (כאשר יש מספר סופי של מקרים שדורשים העברות ושל העברות).  
אם נבצע מעברים נעדן את המצביע בין המתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ של השלשה שממוינת לפי כוחות ולהפך (מצביע דו כיווני). נעדן את המצביע ממתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ הכוחות של הקבוצה ולהפך (מצביע דו כיווני) בתוספת סיבוכיות של  $O(1)$  לאחר שמצאנו את הצמתים המתאימים.  
לאחר כל הסרה (או מעבר בין השלשות) נקרא לפונקציית עזר `updateMinMax` שמתבצעת ב $O(\log n)$  לפי המפורט בפונקציה 9 ותעדן עד 3 ערכים מקסימליים ומינימליים בעצי השלשות. נעדן את המצביע שיש מהשחקן לקבוצה ממנה הוסר להצביע על `nullptr` ב $O(1)$ .  
נקטין את מספר השחקנים בקבוצה ב $O(1)$ .  
סך הכל פעולה זו מתבצעת ב $O(\log n + \log m)$ .

11. `update_contestant_strength` - תחילה נחפש איבר עם `contentantID` בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב $O(\log n)$ . אם לא קיים איבר עם מזהה זה נחזיר `FAILURE`. אחרת, נבדוק אם לאחר תוספת `change` לכוח שלו הכוח יהיה שלילי ב $O(1)$ . אם זה יהיה שלילי נחזיר `FAILURE`. אחרת נחשב את הכוח החדש ככוח שלו ועוד `change` ב $O(1)$  ונעדן את הכוח החדש שלו בעץ המתחרים הכללי הממויין לפי ID ב $O(1)$  (לאחר שכבר מצאנו אותו, כאשר אין צורך להוציא אותו מעץ זה ולהכניס אותו מחדש כי העץ ממויין לפי ID שלא משתנה). כעת, נעדן את שאר הצמתים שמתאימים למתחרה זה בעצים האחרים.  
באמצעות המצביעים שיש למתחרה לקבוצות שהוא שייך אליהן (כאשר נתון שהוא משתתף בכלל היותר 3 קבוצות), נוכל למצוא את הקבוצות בהן הוא משתתף בתוספת סיבוכיות של  $O(1)$ . בכל הקבוצות בהן הוא משתתף נחפש באיזה אחד מעצי השלשות של המתחרים בקבוצה שממוינים לפי ID הוא נמצא ע"י חיפוש שלו בכל שלשה כזאת ב $O(\log n)$ , כך נמצא באיזה שלשה הוא ונוכל לגשת לשלשה הממוינת לפי כוחות המתאימה לשלשה שממוינת לפי ID בה הוא נמצא בתוספת סיבוכיות

של  $O(1)$  ע"י גישה לשלשה המתאימה באמצעות מצביע, נסיר אותו משלשה זו וגם מעץ הכוחות של השחקנים בקבוצה (נמצא אותו שם ב  $O(1)$  באמצעות המצביע מהמתחרה בשלשה של ID שמצאנו למתחרה עם אותו ID בעץ הכוחות של הקבוצה), פעולות אלו מבוצעות ב  $O(\log n)$  ולכל עץ נכניס חוליה חדשה עבור המתחרה עם הכוח המעודכן ב  $O(\log n)$ .

נעדכן את המצביע ממתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ של השלשה שממוינת לפי כוחות ולהפך (מצביע דו כיווני) בתוספת סיבוכיות של  $O(1)$  לאחר שמצאנו את הצמתים המתאימים. נעדכן את המצביע ממתחרה בעץ של השלשה שממויין לפי ID להצביע לאותו מתחרה בעץ הכוחות של הקבוצה ולהפך (מצביע דו כיווני) בתוספת סיבוכיות של  $O(1)$  לאחר שמצאנו את הצמתים המתאימים.

לאחר עדכונים אלו בכל קבוצה בה הוא משתתף נקרא לפונקציית עזר  $updateMinMax$  שמתבצעת ב  $O(\log n)$  לפי המפורט בפונקציה 9 ותעדכן עד 3 ערכים מקסימליים ומינימליים בעצי השלשות. סך הכל סיבוכיות פעולה זו היא  $O(\log n + \log m)$ .

12.  $get\_strength$  - תחילה נחפש איבר עם  $constantID$  בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב  $O(\log n)$ . אם לא קיים איבר עם מזהה זה נחזיר  $FAILURE$ . אחרת, נחזיר את ערך השדה ששומר את הכוח של המתחרה בתוספת סיבוכיות של  $O(1)$  (כי הערך שמור כשדה של מתחרה). סך הכל פעולה זו מתבצעת ב  $O(\log n)$ .

13.  $get\_medals$  - תחילה נחפש איבר עם  $countryID$  בעץ השחקנים הכללי הממויין לפי ID, חיפוש זה מתבצע ב  $O(\log k)$ . אם לא קיים איבר עם מזהה זה נחזיר  $FAILURE$ . אחרת, נחזיר את ערך השדה ששומר את מספר המדליות של מדינה בתוספת סיבוכיות של  $O(1)$  (כי הערך שמור כשדה של מדינה). סך הכל פעולה זו מתבצעת ב  $O(\log k)$ .

14.  $get\_team\_strength$  - תחילה נחפש איבר עם  $teamID$  בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב  $O(\log m)$ . אם לא קיים איבר עם מזהה זה נחזיר  $FAILURE$ . אחרת, ניגש למדינה אליה היא שייכת בתוספת סיבוכיות של  $O(1)$  באמצעות מצביע ששומר לה למדינה אליה היא שייכת. ניגש לעץ הקבוצות של המדינה ב  $O(1)$  ונחפש את הקבוצה שם ב  $O(\log m)$ . כעת נחשב את מספר השחקנים בקבוצה זו ב  $O(1)$  באמצעות בדיקת גדלי עצי השלשות שממויינות לפי ID (כאשר גדלי השלשות שמור כשדה והגישה לשלשות מהקבוצה שמצאנו היא  $O(1)$ ). אם מספר המשתתפים בנבחרת לא מתחלק ב 3 הכוח של הנבחרת יהיה 0, אחרת נחשב את כוח הקבוצה ב  $O(1)$  באמצעות חיבור ערכי  $max$  של הכוחות של כל עצי השלשות של משתתפים שממויינים לפי כוח כאשר כל שלשה נקבעת לפי השלשה של ה ID. כאשר ערכים אלו ששמורים כשדה בשלשות (ומעודכנים בכל פעם שיש שינוי במשתתפים בקבוצה. למשל בהוספת שחקן לקבוצה, בהסרת שחקן מקבוצה ובעדכון גודל כוח של שחקן) לכן חישוב זה יתבצע ב  $O(1)$ . נשים לב ששמירת עצי שלשות של שחקנים שממויינים לפי ID וכוח מקסימלי שמתאים לשחקנים בכל שלשה יוצרת לנו את הסידור שביקשו בפונקציה זו ומאפשרת לחשב את זה בתוספת סיבוכיות של  $O(1)$ . סך הכל פעולה זו מתבצעת ב  $O(\log m)$ .

15.  $unite\_teams$  – תחילה נחפש איבר עם  $teamID1$  בעץ הקבוצות הכללי הממויין לפי ID ואיבר עם  $teamID2$  בעץ הקבוצות הכללי הממויין לפי ID, פעולות אלו מתבצעות ב  $O(\log m)$ . אם אין כאלו נחזיר  $FAILURE$ . אחרת, נבדוק ב  $O(1)$  אם המדינות עליהן הן מצביעות או הספורט שונה. אם המדינה/הספורט שונה נחזיר  $FAILURE$ . אחרת, נמצא את הקבוצות המשווייכות למדינה ב  $O(\log m)$  באמצעות גישה למדינה עם מצביעים מהקבוצות שמצאנו (כאשר בקבוצות השייכות למדינה שמורות השלשות של ה ID ושל הכוחות ועץ הכוחות של כל קבוצה).

לכל קבוצה נגדיר 3 מערכים (1 לכל שלשה של ID) וכל מערך כזה יכיל את שחקני אותה קבוצה ממויינים לפי ID באמצעות סיוור  $Inorder$ . פעולות אלו יתבצעו ב  $O(n_{Team1ID})$  עבור קבוצה 1 וב  $O(n_{Team2ID})$  לפי ההרצאה. בקבוצה 1 נמזג בין מערכים 1 ו 2 ואז נמזג בין מה שהתקבל עם מערך 3 כך שבכל מיזוג נשמור על הערכים ממויינים (נכניס אותם לפי ערכם מהקטן לגדול). פעולה זו תתבצע ב  $O(n_{Team1ID})$ . באופן דומה נפעל עבור קבוצה 2 ונקבל מערך ממויין לפי ID של השחקנים

בה ב- $O(n_{Team2ID})$ . כעת נמזג בין המערך שהתקבל עבור קבוצה 1 (שמכיל את ערכי כל השלשות שלה) ובין המערך שהתקבל עבור קבוצה 2 (שמכיל את ערכי כל השלשות שלה). מיזוג זה יתבצע ב- $O(n_{Team1ID} + n_{Team2ID})$  ונבצע את המיזוג כך שהערכים יהיו ממויינים. נוכל לדעת מה גודל המערך המאוחד בסיבוכיות זו ע"י סכימת מספר השחקנים בקבוצות ולפי מספר השחקנים הכולל ב-2 הקבוצות נוכל לדעת מה הגדלים החדשים של השלשות כך שהתנאי ב- $(*)$  בפונקציה 9 ישמר ונעדכן את גדלי השלשות. נעבור על האיברים לפי הסדר ונעדכן באיזה שלשה הם לפי גדלי השלשות שחישבנו. פעולה זו תתבצע ב- $O(n_{Team1ID} + n_{Team2ID})$ . ניצור עץ AVL לכל שלשה ב- $O(n_{Team1ID} + n_{Team2ID})$  לפי ההרצאה ונעדכן שהם יהיו עצי השלשות החדשים לID. ניתן לבצע *inorder* לשלשות אלו ולעדכן את המצביעים מהשלשות של הכוחות אליהם (בשלשות הID) ומעץ הכוחות של הקבוצה אליהם. באופן דומה נבצע *inorder* לעץ כוחות של קבוצה 1 ב- $O(n_{Team1ID})$  ולעץ כוחות של קבוצה 2 ב- $O(n_{Team2ID})$  ונמזג ביניהם ב- $O(n_{Team1ID} + n_{Team2ID})$ . ניצור 3 מערכים (1 לכל שלשה של כוחות) ונעבור על מערך הכוחות הממוזג של הקבוצות וכל איברים נשים במערך של השלשה המתאימה לו לפי מצביע לשחקן בשלשה של הID ובדיקה באיזה שלשה הוא ב- $O(1)$  כאשר מעבר על המערך הממוזג והכנסת האיברים לשלושת המערכים יתבצע ב- $O(n_{Team1ID} + n_{Team2ID})$ . ניצור עץ AVL לכל שלשה ב- $O(n_{Team1ID} + n_{Team2ID})$  לפי ההרצאה כאשר האיברים יהיו ממויינים כי עברנו עליהם בסדר של המערך הממוזג של הכוחות של הקבוצות שהיה ממויין. ניתן לבצע *inorder* לכל שלשה ולעדכן את המצביעים של השחקנים בשלשות של הID להצביע על השחקן המתאים להם בשלשה של עץ הכוחות ב- $O(n_{Team1ID} + n_{Team2ID})$ . נעדכן שאלו יהיו העצים החדשים ואת הגדלים שלהם. נעדכן גם את המצביעים משחקנים בשלשה של ID לשחקנים בעץ כוחות הממוזג של 2 הקבוצות ב- $O(n_{Team1ID} + n_{Team2ID})$ . במעבר על מערכי השחקנים ניתן לעדכן עבור השחקנים שהשתתפו ב-*team2* שהמצביע ב-*team2* יצביע *nullptr* כי קבוצה זו תוסר. נסיר את קבוצה 2 מעץ הקבוצות של המדינה אליה היא משוייכת ב- $O(\log m)$  כאשר יש מצביע מהקבוצה למדינה שלה לכן ניתן להגיע למדינה ב- $O(1)$  ונסיר את הקבוצה מעץ הקבוצות ב- $O(\log m)$ . לאחר עדכונים אלו בכל קבוצה בה הוא משתתף נקרא לפונקציית עזר *updateMinMax* שמתבצעת ב- $O(\log(n_{Team1ID} + n_{Team2ID}))$  לפי המפורט בפונקציה 9 ותעדכן עד 3 ערכים מקסימליים ומינימליים בעצי השלשות. כאשר זה בפרט ב- $O(n_{Team1ID} + n_{Team2ID})$ . סך הכל פעולה זו תתבצע ב- $O(\log m + n_{Team1ID} + n_{Team2ID})$ .

16. *play\_match* - תחילה נחפש איבר עם *teamID1* בעץ הקבוצות הכללי הממויין לפי ID ואיבר עם *teamID2* בעץ הקבוצות הכללי הממויין לפי ID, פעולות אלו מתבצעות ב- $O(\log m)$ . אם אחת מהקבוצות לא קיימת נחזיר *FAILURE*. אחרת, נבדוק שהנבחרות משתתפות באותו הספורט ב- $O(1)$  באמצעות שדה *m\_sport* ששמור להן. אם הן לא משתתפות באותו ספורט נחזיר *FAILURE*. אחרת, לכל קבוצה ניגש למדינה שהן משתתפות בה ב- $O(1)$  באמצעות מצביע ששמור למדינה זו ואז ניגש למספר המדליות של המדינה ב- $O(1)$  (כי זה שדה ששמור עבור מדינה), נחשב סכום של מספר המדליות ושל הכוח של הנבחרת שמחושב ב- $O(\log m)$  לפי הפירוט בפעולה 14, לכן החישוב יתבצע ב- $O(\log m)$ . נבדוק עבור איזו מדינה הערך שחושב יותר גדול ב- $O(1)$ . אם הניקוד שווה המשחק יסתיים בתיקו. אחרת, נוסיף לנבחרת שניצחה מדליה ב- $O(1)$  באמצעות גישה למדינה שהיא משתתפת בה ב- $O(1)$  באמצעות מצביע ששמור למדינה זו ואז גישה למספר המדליות של המדינה ב- $O(1)$  (כי זה שדה ששמור עבור מדינה) והוספת 1 למספר זה. סך הכל פעולה זו תתבצע ב- $O(\log k + \log m)$ .

17. *austerity\_measures* - תחילה נחפש איבר עם *teamID* בעץ הקבוצות הכללי הממויין לפי ID, חיפוש זה מתבצע ב- $O(\log m)$ . אם לא קיים איבר עם מזהה זה נחזיר *FAILURE*. אחרת, ניגש

למדינה אליה היא שייכת בתוספת סיבוכיות  $O(1)$  באמצעות מצביע ששמור לה למדינה אליה היא שייכת. ניגש לעץ הקבוצות של המדינה ב- $O(1)$  ונחפש את הקבוצה שם ב- $O(\log m)$ . כעת נחשב את מספר השחקנים בקבוצה זו ב- $O(1)$  באמצעות בדיקת גדלי עצי השלשות שממויינות לפי ID (כאשר גדלי השלשות שמור כשדה). אם מספר המשתתפים בבחירת קטן מ-3 נחזיר *FAILURE*. אחרת, נפריד למספר סופי של מקרים שיכולים להתקיים לאחר הסרת 3 שחקנים כאשר בכל מקרה נחשב ב- $O(1)$  את הערך שצריך להחזיר, לכן עבור מספר סופי של מקרים כאלו תוספת הסיבוכיות תהיה  $O(1)$ . כאשר אם לפני הסרת השחקנים בכל שלשה היו  $S$  שחקנים, אז לאחר הסרת 3 שחקנים בכל שלשה אמורים להישאר  $S - 1$  שחקנים.

נפריד למקרים לפי מספר מתחרים שיוסרו מכל שלושה (כאשר מספר השחקנים הוא לפחות 9 ועבור מספר קטן יותר נחשב בנפרד). מקרים אלו מכסים את כל האפשרויות, כי לכל אפשרות אופטימלית של הסרת השחקנים, אפשר לדמות אותה ולהעביר את השחקנים למיקומם האופטימלי (השיג) על ידי הסרה של שלושה שחקנים עם כוח מינימלי ובכך לקבל כוח קבוצתי זהה לכוח הקבוצתי האופטימלי (ייתכן שהסידור שונה). השחקנים בעלי הכוח המינימלי לא משפיעים על הכוח הקבוצתי כי הכוח הקבוצתי מחושב לפי הערכים המקסימליים בכל שלושה שתתקבל במצבים אלו ולכן הסרתם משמשת רק בשביל להזיז שחקנים אחרים. המקרים הם –

- הסרת מתחרה 1 מכל שלשה – במקרה זה הכוח המקסימלי של הנבחרת יחושב לפי *get\_team\_strength* (לפי הגדרת הפונקציה, כאשר ישארו השחקנים עם הכוח המקסימלי) ב- $O(\log m)$ .

- הסרת 2 שחקנים מהשלשה השמאלית ושחקן אחד מהשלשה הימנית – במצב זה הכוח המקסימלי של השלשה השמאלית החדשה יחושב לפי המקסימום מבין הכוח המקסימלי של השלשה השמאלית לפני ההסרה ומבין הכוח של השחקן עם *minID1* של השלשה האמצעית לפני ההסרה. הכוח המקסימלי של השלשה האמצעית יהיה הכוח המקסימלי השני בגודלו של השלשה האמצעית לפני ההסרה אם ה-ID שלו שווה ל-ID של *minID1* מהשלשה האמצעית לפני ההסרה, אחרת הכוח המקסימלי של השלשה האמצעית לפני ההסרה יהיה הכוח המקסימלי הראשון של השלשה האמצעית לפני ההסרה. הכוח המקסימלי של השלשה הימנית יהיה הכוח המקסימלי של השלשה הימנית לפני ההסרה.

- הסרת 2 מתחרים מהשלשה האמצעית ומתחרה 1 מהשלשה השמאלית – במצב זה הכוח המקסימלי של השלשה השמאלית החדשה יהיה הכוח המקסימלי של השלשה השמאלית לפני ההסרה. הכוח המקסימלי של השלשה האמצעית יהיה הכוח המקסימלי של השלשה האמצעית בין הכוח המקסימלי של השלשה האמצעית לפני ההסרה ובין הכוח של *minID1* של השלשה הימנית לפני ההסרה. הכוח של השלשה הימנית יהיה הכוח המקסימלי השני בגודלו של השלשה הימנית לפני ההסרה אם ה-ID של הכוח המקסימלי של השלשה הימנית לפני ההסרה שווה ל-ID של *minID1* של השלשה הימנית לפני ההסרה.

- הסרת 2 מתחרים מהשלשה האמצעית ומתחרה 1 מהשלשה הימנית – במקרה זה אם ה-ID של *maxID1* של השלשה השמאלית לפני ההסרה שווה ל-ID של השחקן עם הכוח המקסימלי של השלשה השמאלית לפני ההסרה אז הכוח המקסימלי של השלשה השמאלית החדשה יהיה הכוח המקסימלי השני בגודלו שהיה בשלשה זו לפני ההסרה, אחרת הוא יהיה הכוח המקסימלי שהיה בשלשה זו לפני ההסרה. הכוח המקסימלי בשלשה האמצעית החדשה יהיה הכוח של *maxID1* מהשלשה השמאלית לפני ההסרה ובין הכוח המקסימלי של השלשה האמצעית לפני ההסרה. הכוח המקסימלי של השלשה הימנית החדשה יהיה שווה לכוח המקסימלי של השלשה הימנית לפני ההסרה.

- הסרת 2 מתחרים מהשלשה הימנית ומתחרה 1 מהשלשה השמאלית – במצב זה הכוח המקסימלי בשלשה השמאלית החדשה יהיה הכוח המקסימלי שהיה בשלשה זו לפני ההסרה. אם ה-ID של *maxID1* בשלשה האמצעית לפני ההסרה שווה ל-ID של השחקן עם הכוח המקסימלי בשלשה האמצעית לפני ההסרה אז הכוח המקסימלי בשלשה האמצעית החדשה יהיה הכוח המקסימלי השני בגודלו שהיה בשלשה זו לפני ההסרה, אחרת הוא יהיה שווה לכוח המקסימלי שהיה בשלשה לפני ההסרה. הכוח המקסימלי של השלשה הימנית יהיה הכוח המקסימלי בין הכוח של *maxID1* של השלשה האמצעית ושל הכוח המקסימלי של השלשה הימנית לפני ההסרה.

- ישנם גם המקרים הבאים - הסרת 2 מתחרים מהשלשה הימנית ומתחרה 1 מהשלשה האמצעית,



הסרת 2 שחקנים מהשלשה השמאלית ושחקן 1 מהשלשה האמצעית, הסרת 3 שחקנים מהשלשה השמאלית, הסרת 3 שחקנים מהשלשה האמצעית והסרת 3 שחקנים מהשלשה הימנית בהם באופן דומה יש מספר סופי של מצבים שיכולים להתקיים וחשובים של הערך המקסימלי של כל שלשה חדשה באמצעות ערכי  $min$  ו- $max$  ששמורים לנו.

ערכי  $min$  ו- $max$  בהם השתמשנו בכל מקרה מתקבלים ב- $O(1)$  כי הם שמורים בשלוש ומעודכנים לאורך הפעולות.

בכל מקרה נחשב את הכוח המקסימלי כסכום הכוחות המקסימיים של השלוש  $O(1)$  כי יש מספר סופי של מקרים ומספר סופי של שלשות ונחזיר את כמות הכוח המקסימלית שמתקבלת כ- $max$  מכל המקרים ב- $O(1)$  ע"י מספר סופי של השוואות כי מספר המקרים סופי.

אם יש 3 שחקנים אז לאחר הסרת 3 שחקנים לא ישארו שחקנים בקבוצה לכן נחזיר 0. (בדיקה זו והחזרת הערך מתבצעות ב- $O(1)$ ).

אם יש 6 שחקנים אז ניתן להסיר לכל היותר 2 מכל קבוצה ובאופן דומה יש מספר סופי של מקרים של מאיפה יוסרו שחקנים וכמה ובכל אחד מהמקרים נחשב באופן דומה איך היו נראות השלוש לאחר הסרת השחקנים באמצעות שימוש בערכי  $max$  ו- $min$  ששמרנו עבור השלוש, לכן גם חישוב זה יתבצע ב- $O(1)$  ונחזיר את כמות הכוח המקסימלית שמתקבלת מכל המקרים ב- $O(1)$  ע"י מספר סופי של השוואות כי מספר המקרים סופי.

סך הכל סיבוכיות הזמן בפעולה זו היא  $O(\log m)$ .

סיבוכיות מקום - מבנה הנתונים כולל עץ מדינות שממויין לפי ID (של מדינה) בו יהיו לכל היותר  $k$  איברים, עץ קבוצות שממויין לפי ID (של קבוצה) בו יהיו לכל היותר  $m$  איברים ועץ מתחרים שממויין לפי ID של מתחרים בו יהיו לכל היותר  $n$  איברים. בנוסף, לכל קבוצה יהיו 6 עצים כך ש-3 עצים מתוכם יהיו ממויינים לפי ID של השחקנים בקבוצה ויהיו מחולקים לשלוש ושלושת העצים האחרים יהיו עצים הממויינים לפי כוחות השחקנים של כל שלשה (כלומר לכל עץ שממויין לפי ID ומתאר שלשה של מתחרים יהיה עץ שממויין לפי כוחות ומתאים למתחרים של אותה שלשה) ולכל קבוצה יהיה עץ שממויין לפי כוחות כל השחקנים שבקבוצה. סך האיברים מכל השלוש של ID מכל הקבוצות הוא לכל היותר  $O(n)$  כי יש לכל היותר  $O(n)$  שחקנים בכל המערכת. באופן דומה סך האיברים מכל השלוש שממויינות לפי כוחות (כאשר כל שלשה של כוחות מתאימה לשלשה של ID) הוא  $O(n)$  וגם סך האיברים מעצי הכוחות הכלליים של כל קבוצה הוא  $O(n)$  (כי יש לכל היותר  $O(n)$  שחקנים בכל המערכת).

לכן, סך הכל סיבוכיות המקום היא  $O(n + k + m)$ .