**תרגיל בית 1 – יבש**

World Cup –>{\*team avl tree, \*player avl tree by id, \*player avl tree by score, player count, \*top scorer, \*avl tree active teams}

Team ->{\*player avl tree by id, \*player avl tree by score, points, goalie cnt, id, player count, strength, games played, \*top scorer}

Player->{id, goals, games played, is goalie, cards, \*team, \*prev player, \*next player, \*prev player in team, \*next player in team}

**world\_cup\_t**

**תיאור:**

מאתחל את מבני הנתונים.

**~world\_cup\_t**

**תיאור:**

משחרר את הכל.

**add\_team**

**תיאור:**

יוצר קבוצה חדשה עם הפרמטרים שניתנו ומכניס את הקבוצה לעץ הקבוצות שהוא עץ AVL כפי שנלמד בהרצאה (עם פונקציית השוואה לפי id בסיבוכיות O(1)) בסיבוכיות של O(log(k)). נאתחל את כוח למספר הנקודות שניתנו ואת מספר המשחקים ל-0.

**remove\_team**

**תיאור:** קודם מחפש את הקבוצה בעץ הקבוצות עם סיבוכיות של O(log(k)), אז בודק אם יש שחקנים בקבוצה, אם אין שחקנים משחרר את הקבוצה מהזיכרון ומסדר את העץ כמו שנלמד בהרצאה ב-O(log(k)).

**add\_player**

**תיאור:**

-קודם נחפש את הקבוצה שצריך להוסיף לה את שחקן ב-O(log(k)).

-נחסיר מכמות המשחקים של אותו שחקן את כמות המשחקים של הקבוצה באותו הרגע.

-אחר כך ניצור את השחקן עם הפוינטר של הקבוצה שהגענו אליה.

-אז נכניס את השחקן לקבוצה לעץ של השחקנים לפי ID בסיבוכיות של O(log(nteam)) כמו שלמדנו בהרצאה.

-לאחר מכן נוסיף לעץ שחקנים לפי ניקוד בקבוצה את השחקן כמו שנלמד בהרצאה. לפני הגילגולים!!!!! נגדיר את השחקן הקודם\הבא בקבוצה של השחקן שהוספנו להיות האבא של השחקן (לוגית זה עובד נראה לי) ונעדכן גם את השחקן הקודם\הבא של השחקנים האחרים בקבוצה שהושפעו מההכנסה.

-בנוסף נעדכן את מספר השחקנים בקבוצה ואת מספר השחקנים הכולל ואת הכוח של הקבוצה בכך שנוסיף את כמות הגולים שהכניס ונחסיר את כמות הכרטיסים שקיבל.

-אם השחקן שוער נוסיף 1 למספר השוערים בקבוצה.

-נבדוק אם הקבוצה הפכה להיות פעילה מלא פעילה, אם הפכה לפעילה נכניס אותה לעץ של הקבוצות הפעילות לפי הID של הקבוצה.

-בנוסף נבדוק אם השחקן הכניס יותר גולים מהמכניס הראשי של הקבוצה ונעדכן את המכניס הראשי בהתאם.

-לאחר מכן נוסיף לעץ שחקנים לפי ID את השחקן כמו שנלמד בהרצאה.

-לאחר מכן נוסיף לעץ שחקנים לפי ניקוד את השחקן כמו שנלמד בהרצאה. לאחר ההכנסה אבל לפני הגלגולים!!!!! נגדיר את השחקן הקודם\הבא של השחקן שהוספנו להיות האבא של השחקן (לוגית זה עובד נראה לי) ונעדכן גם את השחקן הקודם\הבא של השחקנים האחרים שהושפעו מההכנסה.

-בנוסף נבדוק אם השחקן הכניס יותר גולים מהמכניס הראשי של כולם ונעדכן את המכניס הראשי בהתאם.

**remove\_player**

**תיאור:**

-נחפש את השחקן בעץ של השחקנים לפי ID ונשמור את המצביע לשחקן.

-לאחר מכן נסיר את השחקן מעץ השחקנים לפי ID בקבוצה שהוא נמצא בה ומעץ השחקנים לפי ניקוד בקבוצה.

-נעדכן את המכניס הראשי בקבוצה עם המצביע לשחקן הקודם בניקוד בקבוצה ששמור בשחקן.

-נעדכן את מספר השחקנים בקבוצה ואת כוח הקבוצה בכך שנוסיף לו את מספר הכרטיסים שקיבל ונחסיר את מספר הגולים שהכניס.

-אם השחקן שוער נעדכן את כמות השוערים.

-אם הקבוצה הפכה להיות לא פעילה נסיר אותה מעץ הקבוצות הפעילות.

-לאחר מכן נסיר את השחקן מעץ השחקנים לפי ID ומעץ השחקנים לפי ניקוד.

-נעדכן את המכניס הראשי עם המצביע לשחקן הקודם בניקוד ששמור בשחקן.

-נעדכן את השחקנים שצמודים אליו בניקוד.

-בסוף נמחק את השחקן עצמו באמצעות הפוינטר ששמרנו.

**update\_player\_stats**

**תיאור:**

נחפש את השחקן בעץ השחקנים לפי ID ונוסיף לו את הנתונים שקיבלנו.

בעיקרון הוצאה והכנסה של השחקן רק בלי החיפוש של הקבוצה.

**play\_match**

**תיאור:**

נחפש את 2 הקבוצות ונשווה בין הכוחות שלהם ונעדכן את הנקודות של הקבוצות בהתאם לתוצאה. בנוסף נוסיף לכל קבוצה 1 למספר המשחקים ששיחקו.

**get\_num\_played\_games**

**תיאור:**

נחפש את השחקן בעץ השחקנים לפי ID ונחזיר את מספר המשחקים ששיחק ועוד מספר המשחקים ששיחקה הקבוצה שלו.

**get\_team\_points**

**תיאור:**

נחפש את הקבוצה בעץ הקבוצות ונחזיר את מספר הנקודות שלו.

**unite\_teams**

**תיאור:**

-קודם נחפש את 2 הקבוצות בעץ הקבוצות.

-לאחר מכן נעדכן את מספר המשחקים של כל שחקן בכך שנוסיף לו את כמות המשחקים של הקבוצה. לכל קבוצה.

-כעת נשמור את סכום הנקודות, סכום השוערים וסכום הכוח של 2 הקבוצות.

-ניצור קבוצה חדשה עם הID, הניקוד ומספר השחקנים המתאים.

נכניס את העץ שחקנים לפי ID של כל קבוצה למערך ב-in order כדי לשמור על הסדר.

תוך כדי ההכנסה נעדכן את המצביע לקבוצה של כל שחקן.

נכניס את העץ שחקנים לפי ניקוד של כל קבוצה למערך ב-in order כדי לשמור על הסדר.

נמזג בין המערכים לפי ID למערך לפי ID אחד מסודר.

נמזג בין המערכים לפי ניקוד למערך לפי ניקוד אחד מסודר.

נסדר את השחקנים כך שהשחקן הקודם\הבא לפי קבוצה שלהם יהיה מתוקן לפי הסדר של המערך לפי ניקוד.

נמחק את 2 הקבוצות מהעץ של הקבוצות.

אם הקבוצות פעילות נמחק אותם מעץ הקבוצות הפעילות

נכניס את הקבוצה שיצרנו לעץ הקבוצות.

אם הקבוצה שיצרנו פעילה נוסיף אותה מעץ הקבוצות.

**get\_top\_scorer**

**תיאור:**

אם ביקשו מכולם נחזיר את הID המתאים מהמצביע השמור.

אם ביקשו מקבוצה, נחפש את הקבוצה בעץ הקבוצות ונחזיר את המכניס הראשי שלה.

**get\_all\_players\_count**

**תיאור:**

אם ביקשו מכולם נחזיר את המשתנה שסופר את מספר השחקנים.

אם ביקשו מקבוצה, נחפש את הקבוצה ונחזיר את מספר השחקנים ששמור אצלה.

**get\_all\_players**

**תיאור:**

נעבור לפי הסדר על העץ המתאים ונכניס אותו למערך פלט.

**get\_closest\_player**

**תיאור:**

נחפש את הקבוצה של השחקן (אם רוצים לפי קבוצה).

נחפש את השחקן הרצוי.

נבדוק למי הוא יותר קרוב מבין שני השחקנים ששמורים אצלו.

נחזיר את השחקן שמתאים.

**knockout\_winner**

**תיאור:**

נכניס את העץ של הקבוצות הפעילות לתוך מערך דו מימדי לפי הסדר ב-in order כאשר בתא הראשון הID של הקבוצה ובתא השני הכוח שלה.

נמשיך את מהלך המשחק ונחזיר את המנצח.

**sibuchiut\_makom**

**תיאור:**

**zain batahat**