

קורס: מבני נתונים 1 - 234218

חלק יבש לרטוב 1

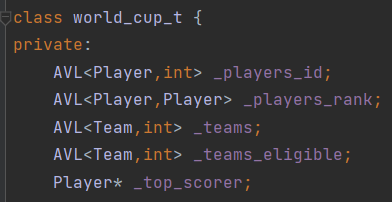
מגישות:

1. אמילי דרור (324934082)
2. בראה חליפה (322777236)

תאריך הגשה: 13.12.2022

**תיאור מבנה הניתונים**

מחלקת world\_cup\_t

במחלקת קיימים שני עצי AVL למחלקות Team ושני עצי AVL נוספים למחלקת Player שמתוארות בהמשך, בנוסף יש שדה \_top\_scorer שמצביע לשחקן.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Team

במחלקת קיימים שדות של int ושני עצי AVL למחלקת Player.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת Player

*מחלקה מתארת שחקן, כל שחקן מחזיק מצביע לקבוצה שמשחק בה.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

מחלקת AVL

*תיאורי קצר: מחלקה זו גנרית במפתחות וגם במידע שצריך להחזיק*

*שדות: מצביע לשורש, גודל העץ.*

*חיפוש, הוספה והוצאה – לפי האלגוריתמים שנלמדו בכיתה .*

*סיורים בעץ – לפי האלגוריתמים שנלמדו בכיתה .*

*איחוד שני עצים – בעזרת סיור* inorder *ניצור שני מערכים ממוינים של איברי שני העצים, נאחד את שני המערכים בעזרת פונקצית* merge *פשוטה וניעזר בפונקציה אחרת שמבצעת את ההפך מסיור* inorder *כדי ליצור עץ ונחזיר אותו, הסיבוכיות של הפתרון היא , כאשר הוא גודל שני העצים ביחד.*

*פונקציות מקסימום ומינימום – נחזיר את האיבר בעל המפתח הכי גדול או קטן על ידי החזרת העלה הימנית ביותר או השמאלית ביותר .*

**תיאור המתודות של world\_cup\_t:**

constructor

*בבנאי קוראים לבנאי של כל עץ, נבחר שני עצים של* Team *ו-* Player *להיות אחרים על שחרור הזיכרון ואת השאר רק מחזיקים מצביעים אך לא משחררים אותם. בנוסף נאתחל* \_top\_scorer *ל-* nullptr.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

destructor

*הורס דיפולטי, כל עץ קורא להורס שלו, שני העצים האחראיים על שחרור הזיכרון משחררים את הזיכרון של השחקנים והקבוצות.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_team(int teamId, int points)

*תיאור: בהינתן המספר המזהה* teamId *ניצור קבוצה חדשה ונוסיף אותה לעץ* world\_cup\_t::teams.

*סיבוכיות: הוספה לעץ* AVL *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType remove\_team(int teamId)

*תיאור: בהינתן המספר המזהה* teamId *נוציא את הקבוצה מעץ* world\_cup\_t::teams, *נבדוק באמצעות מתודה אם יש לה 11 שחקנים ושוער אחד לפחות ואם כן נוציא אותה מעץ* world\_cup\_t::teams\_eligible.

*סיבוכיות: הוצאה מעץ* AVL *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType add\_player(int playerId, int teamId, int gamesPlayed, int goals, int cards, bool goalKeeper)

*תיאור: בהינתן הקלט הנ"ל ניצור שחקן חדש ונוסיף אותה לעץ* world\_cup\_t::players\_id *ולעץ* world\_cup\_t::players\_rank, *אחר כך נחפש את הקבוצה בעלת המזהה* teamId *ונוסיף השחקן לשני העצים בתוכה.*

*סיבוכיות: הוספה לעץ* world\_cup\_t:: players\_id *עולה , הוספה לעץ* Team:: players\_id *עולה , חיפוש בעץ* world\_cup\_t::teams *עולה* *, לכן מכיוון ש- אזי הסיבוכיות היא .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType remove\_player(int playerId)

*תיאור: נוציא השחקן בעל המזהה* playerId *משני עצי* world\_cup\_t::players, *בנוסף באמצעות הרפרנס שהשחקן מחזיק לקבוצה שלו נוציא אותו מעץ* Team::players*.*

*סיבוכיות: 3 הוצאות מעצי* **AVL** *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType update\_player\_stats(int playerId, int gamesPlayed, int scoredGoals, int cardsReceived)

*תיאור: נחפש* playerId *בעץ* world\_cup\_t::players *ונעדכן את השחקן באמצעות מתודה ממחלקת* Player*.*

*סיבוכיות: העדכון יהיה ולכן סה"כ .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType play\_match(int teamId1, int teamId2)

*תיאור: נחפש את שתי הקבוצות בעץ* world\_cup\_t::teams\_eligible*, במידה ואחת מהן לא קיימת תיזרק חריגה ויוחזר ערך מתאים, אחר כך נפעיל מתודה על שתי הקבוצות כדי לסמלץ משחק. ונוסיף לשדה* \_games\_to\_add *אחד, שדה זה מייצג כמה משחקים שחקה קבוצה זה וכמה צריך להוסיף לכל שחקן שנמצא בה.*

*סיבוכיות: חיפוש בעץ* world\_cup\_t:: teams\_eligible *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t <int> get\_num\_played\_games(int playerId)

*תיאור: נחפש* playerId *בעץ* world\_cup\_t::players\_id *ונחזיר את הערך של השדה המתאים + הערך משדה* \_games\_to\_add*, כאשר מוסיפים שחקן נחסיר מהשדה שלו* \_games\_to\_add *כדי להחזיר תשובה נכונה בקריאה לשחקן הזה, בפעולת* unite\_teams *נוסיף לכל שחקן את* \_games\_to\_add *ונאפס השדה של הקבוצה החדשה עם עמידה בדרישת הסיבוכיות.*

*סיבוכיות: חיפוש בעץ* world\_cup\_t::players *עולה*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t <int> get\_team\_points(int teamId)

*תיאור: נחפש* teamId *בעץ* world\_cup\_t::teams *ונחזיר את הערך של השדה המתאים.*

*סיבוכיות: חיפוש בעץ* world\_cup\_t::teams *.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType unite\_teams(int teamId1, int teamId2, int newTeamId)

*תיאור: נחפש את שתי הקבוצות בעץ* world\_cup\_t::teams*, ניצור קבוצה חדשה ונוסיף אליה את השדות של שתי הקבוצות האחרות, נאחד את שני העצים בקבוצות הישנות ונוסיף את העץ החדש לקבוצה החדשה. בסוף נוציא את הקבוצות הישנות מעץ* world\_cup\_t::teams *ונוסיף אליו את הקבוצה החדשה.*

*סיבוכיות: חיפוש, הוספה והוצאה עולים , יחד עם האיחוד .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t < int > get\_top\_scorer(int teamId)

*תיאור: נשמור שדה שמתאר את* top\_scorer *במחלקת* world\_cup\_t *ומחלקת* Team*. כאשר מכניסים, מוציאים או מעדקנים שחקן נשים לב לעדכן את השדות האלה.*

*סיבוכיות: אם 0 >* teamId *אז . אחרת צריך לחפש את המזהה לכן . במקרה שצריך לעדכן את השדות נשתמש בפונקצית* AVL::max() *במקרה הכי גרוע שעולה וזאת מכוסה על ידי כל אחת מהפעולות הנ"ל.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t < int > get\_all\_players\_count(int teamId)

*תיאור: אם 0 >* teamId *נחזיר את גודל עץ* world\_cup\_t::players*, אחרת נחפש קודם את הקבוצה ונחזיר את גודל עץ* Team::players*, גודל העץ נשמר כשדה.*

*סיבוכיות: אם 0 >* teamId *אז . אחרת צריך לחפש את המזהה לכן .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

StatusType get\_all\_players(int teamId, int ∗ const output)

*תיאור: בהתאם לסימן* teamId *נבצע סיור* inorder*על* *עץ* world\_cup\_t::players *או עץ* Team::players *ונשמור את המידע* *בבמערך הנתון.*

*סיבוכיות: אם 0 >* teamId *אז . אחרת צריך לחפש את המזהה לכן .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t <int> get\_closest\_player(int playerId, int teamId)

*תיאור: נשמור שני שדות במחלקת* Player*, כאשר מכניסים, מוציאים או מעדכנים שחקן ננצל שפעולות אלה עולות ונעדכן שדות אלה לשחקן עצמו ולשני השחקנים ששדות אלה מצביעים עליהם. עכשיו נחפש את הקבוצה הנתונה ונחפש בעץ שהיא מחזיקה את השחקן, נשווה בין שני השדות ומחזיר את הקרוב ביותר.*

*סיבוכיות: חיפוש בעץ הקבוצות , חיפוש בעץ השחקנים בתוך הקבוצה , השוואה לכן .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

output\_t <int> knockout\_winner(int minTeamId, int maxTeamId)

*תיאור: נבצע סיור* inorder *על עץ* world\_cup\_t:: teams\_eligible *כך שיחזיר את הקבוצות בעלות מזהה בתחום הנתון, אחר כך נבצע בכל שלב את המתואר בשאלה.*

*סיבוכיות: סיור* inorder *עולה , בכל שלב מספר הקבוצות קטן פי 2 והכל נעצר כשנשארת קבוצה אחת לכן יש שלבים ובכל שלב מבצעים פעולות לכן מכיוון ש- הסיבוכיות היא .*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*