* **כמתים**
  + n – כמות הגלדיאטורים הכוללת.
  + k – כמות מאמני הגלדיאטורים הכוללת.
* **תיאור מבנה הנתונים**
  + עץ המאמנים: עץ splay המכיל מאמני גלדיאטורים. כל איבר בעץ מייצג מאמן גלדיאטורים בודד ומכיל שדה עבור מזהה המאמן (Trainer’s ID), שדה המכיל את מספר הגלדיאטורים הרשומים תחת המאמן () ושדה המכיל מצביע ל"עץ הגלדיאטורים הקטן" שלו. לכל מאמן יש מספר (אי שלילי) של גלדיאטורים אותם הוא ורק הוא מאמן. לכן, גלדיאטור הנמצא תחת אחד המאמנים לא יופיע תחת מאמן אחר. על כן, לכל מאמן יהיה את "עץ הגלדיאטורים הקטן" שלו והוא יחזיק מצביע אליו על מנת לגשת אל הלוחמים אותם הוא מאמן בקלות. עץ המאמנים ממויין לפי המזהה של המאמנים (Trainer’s ID) וכך נוכל למצוא מאמן לפי המזהה שלו בסיבוכיות זמן לוגריתמית (אף על פי שזה לא נדרש). עץ המאמנים מכיל k צמתים כמספר מאמני הגלדיאטורים ועל כן סיבוכיות המקום שלו היא מסדר גודל של .
  + עץ הגלדיאטורים הקטן: עץ splay המשמש כשדה מאפיין עבור מאמן גלדיאטורים. כמו שצויין לעיל, היות וכל גלדיאטור משויך למאמן יחיד, ולכל מאמן יש אחריות על מספר אי שלילי של גלדיאטורים, הגלדיאטורים של כל מאמן יוחזקו ב"עץ הגלדיאטורים הקטן" של המאמן ובעץ המאמנים לכל מאמן יהיה מצביע לעץ הגלדיאטורים שלו. איבר בודד ב"עץ הגלדיאטורים הקטן" מאפיין גלדיאטור יחיד וייחודי ומכיל שדה עבור מזהה הגלדיאטור (Gladiator’s ID), שדה עבור מזהה המאמן שאחראי עליו (Trainer’s ID), רמה (Level), מצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים לפי מזהה" ומצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים לפי רמה" על מנת להקל על פעולות הדורשות עדכון פרטי הגלדיאטור. עץ הגלדיאטורים הקטן ממויין לפי הרמה של הגלדיאטורים בסדר יורד וכך נוכל בקלות לקבל מדרג לפי רמה של הגלדיאטורים המשוייכים למאמן יחיד. במידה וקיימים שני גלדיאטורים באותה רמה, הם ימויינו במיון משני לפי מזהה הגלדיאטור בסדר עולה. עץ הגלדיאטורים הקטן מכיל צמתים כמספר הגלדיאטורים המשוייכים למאמן. היות וכל גלדיאטור משוייך למאמן יחיד, ובהתחשב בעובדה שלכל מאמן יש עץ גלדיאטורים, סה"כ במבנה הנתונים יהיו k עצי גלדיאטורים קטנים (כמספר המאמנים) המכילים n צמתים בסך הכל (כמספר הגלדיאטורים במערכת). לפיכך, סיבוכיות המקום של עץ הגלדיאטורים הקטן היא מסדר גודל של .
  + עץ הגלדיאטורים לפי מזהה: עץ splay נפרד המחזיק עותק של כל הגלדיאטורים במערכת ממוינים לפי מזהה הגלדיאטור (Gladiator’s ID) בסדר עולה. איבר בודד בעץ זה מאפיין גלדיאטור יחיד וייחודי ומכיל שדה עבור מזהה הגלדיאטור (Gladiator’s ID), שדה עבור מזהה המאמן שאחראי עליו (Trainer’s ID), רמה (Level), מצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים הקטן" ומצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים לפי רמה" על מנת להקל על פעולות הדורשות עדכון פרטי הגלדיאטור. בנוסף, נחזיק מצביע למאמן של הגלדיאטור. היות ועץ זה ממויין לפי המספר המזהה של הגלדיאטורים, הוא מאפשר לנו לגשת ולמצוא גלדיאטור לפי מספרו המזהה בסיבוכיות זמן לוגריתמית. העץ מכיל צמתים כמספר הגלדיאטורים במערכת ועל כן מכיל n צמתים. לפיכך, סיבוכיות המקום של עץ הגלדיאטורים לפי מזהה היא מסדר גודל של .
  + עץ הגלדיאטורים לפי רמה: עץ splay נפרד המחזיק עותק של כל הגלדיאטורים במערכת ממוינים לפי הרמה של הגלדיאטור (Level) בסדר יורד. במידה וקיימים שני גלדיאטורים באותה רמה, הם ימויינו במיון משני לפי מזהה הגלדיאטור בסדר עולה. איבר בודד בעץ זה מאפיין גלדיאטור יחיד וייחודי ומכיל שדה עבור מזהה הגלדיאטור (Gladiator’s ID), שדה עבור מזהה המאמן שאחראי עליו (Trainer’s ID), רמה (Level), מצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים הקטן" ומצביע לאיבר המייצג את אותו הגלדיאטור ב"עץ הגלדיאטורים לפי מזהה" על מנת להקל על פעולות הדורשות עדכון פרטי הגלדיאטור. בנוסף, נחזיק מצביע למאמן של הגלדיאטור. היות ועץ זה ממויין לפי הרמה של הגלדיאטורים, הוא מאפשר לנו לגשת ולמצוא גלדיאטור לפי רמתו בסיבוכיות זמן לוגריתמית או למצוא את הגלדיאטורים בעלי הרמה הכי גבוהה או נמוכה בקלות יתרה. העץ מכיל צמתים כמספר הגלדיאטורים במערכת ועל כן מכיל n צמתים. לפיכך, סיבוכיות המקום של עץ הגלדיאטורים לפי רמה היא מסדר גודל של .
  + משתנים:
* ??
  + נשים לב כי עץ המאמנים דורש סיבוכיות מקום של ואילו עצי הגלדיאטורים הקטנים, עץ הגלדיאטורים לפי מזהה ועץ הגלדיאטורים לפי רמה דורשים סיבוכיות מקום של כל אחד. לכן, סיבוכיות המקום הכוללת הנדרשת עבור מבנה הנתונים היא סדר גודל של .
* **מימוש הפעולות**
  + **void\* Init()**

אתחול מבנה הנתונים. בעת אתחול, יש ליצור את ארבעת המבנים שפורטו לעיל. היות וארבעת המבנים הינם עצים, הם יאותחלו כעצים ריקים ולכן פעולה זו תתבצע בסיבוכיות זמן של .

* + **StatusType AddTrainer(void \*DS, int trainerID)**

הוספת מאמן חדש למערכת עם המזהה trainerID. המאמנים מאוחסנים בעץ המאמנים בלבד ולכן בפעולה זו נצטרך להוסיף איבר חדש לעץ המאמנים. כמו שנלמד, פעולת insert בעץ מסוג splay מתבצעת בסיבוכיות זמן של כאשר k הוא מספר הצמתים הקיים בעץ. היות ומספר הצמתים בעץ המאמנים הוא k, כמספר המאמנים, הוספת מאמן חדש תתבצע בסיבוכיות זמן של ובפרט בסיבוכיות זמן של כנדרש.

* + **StatusType BuyGladiator(void \*DS, int gladiatorID, int trainerID, int level)**

הוספת גלדיאטור חדש בהינתן המזהה שלו, הרמה שלו והמזהה של המאמן אליו הוא ישוייך. כמו שמפורט לעיל, ישנם שלושה מבנים באים אנו שומרים מידע אודות הגלדיאטורים. ראשית, בהינתן המזהה של מאמן הגלדיאטור, נחפש את המאמן שלו בעץ המאמנים. עץ המאמנים ממויין לפי המספרים המזהים של המאמנים ועל כן מציאת המאמן תתבצע בסיבוכיות זמן של . כעת, משמצאנו את המאמן, ניגש אל עץ הגלדיאטורים הקטן שלו. עץ הגלדיאטורים הקטן ממויין לפי רמת הגלדיאטורים, ולכן, בהינתן הרמה של הגלדיאטור, נבצע insert בסיבוכיות זמן של כאשר במקרה ה"גרוע" ישנם n גלדיאטורים תחת המאמן הנתון. נשמור את המצביע לגלדיאטור בעץ זה. כעת, שוב בהינתן רמת הגלדיאטור, ניגש לעץ הגלדיאטורים לפי רמה ונבצע גם שם פעולת insert. היות וגם עץ זה ממויין לפי רמת הגלדיאטורים הפעולה תתבצע בסיבוכיות זמן של מאחר וישנם n גלדיאטורים בעץ זה. נשמור גם את המצביע לגלדיאטור בעץ זה. לבסוף, ניגש לעץ הגלדיאטורים לפי מזהה ונבצע גם שם פעולת insert. עץ זה ממויין לפי המזהה של הגלדיאטורים, ולכן, בהינתן המזהה של הגלדיאטור, פעולה זו תתבצע בסיבוכיות זמן של מאחר וישנם n גלדיאטורים בעץ. נשמור גם את המצביע לגלדיאטור בעץ זה. כעת, הוספנו את הגלדיאטור לשלושת העצים וברשותנו מצביע לגלדיאטור בכל אחד מהם. נעדכן בכל אחד מהשדות הרלוונטיים את המצביעים לשדות האחרים על מנת לשמור על קישוריות נוחה בין מבני הנתונים. במידה והפעולה הצליחה, נגדיל עבור המאמן את ב-1. נשים לב כי סך כל הפעולות בוצעו בסיבוכיות זמן מסדר גודל של ולכן בפרט גם . כמובן שבעת הוספת איבר לעצים נבצע גלגולים כנדרש, אך כמו שנלמד סיבוכיות הזמן של הפעולות לא תשתנה היות והגלגולים ידרשו לכל היותר כאשר h הוא גובה העץ (במקרה שלנו ) ולכן סיבוכיות הזמן תשאר זהה.

* + **StatusType FreeGladiator(void \*DS, int gladiatorID)**

ברשותנו המזהה של הגלדיאטור ולכן נמצא אותו בעץ הגלדיאטורים לפי מזהה. היות ועץ זה ממויין לפי המספרים המזהים של הגלדיאטור, נמצא את מיקום הגלדיאטור בסיבוכיות זמן של . כעת, דרך איבר זה נוכל למצוא מצביעים לגלדיאטור גם בעץ הגלדיאטורים לפי רמה וגם בעץ הגלדיאטורים הקטן (ובפרט גם מצביע לעץ עצמו – מאחר ויש ברשותנו מצביע למאמן של הגלדיאטור, המכיל את העץ). עכשיו כשברשותנו כל המצביעים, בכל אחד מהעצים נבצע פעולת remove וכנלמד פעולה זה לא תשנה את סיבוכיות הזמן הכוללת מאחר ובכל עץ סיבוכיות הזמן שלה לוגריתמית למספר הצמתים (כאשר בכל אחד מעצים אלה מספר הצמתים הינו n). על כן, פעולה זו בוצעה בסיבוכיות זמן מסדר גודל של ובפרט גם בסיבוכיות זמן מסדר גודל של . במידה והפעולה הצליחה, נקטין עבור המאמן את ב-1. כמובן שבעת הוצאת איבר מעץ נבצע גלגולים כנדרש, אך כמו שנלמד סיבוכיות הזמן של הפעולות לא תשתנה היות והגלגולים ידרשו לכל היותר כאשר h הוא גובה העץ (במקרה שלנו ) ולכן סיבוכיות הזמן תשאר זהה.

* + **StatusType LevelUp(void \*DS, int gladiatorID, int levelIncrease)**

ברשותנו המזהה של הגלדיאטור ולכן נמצא אותו בעץ הגלדיאטורים לפי מזהה בסיבוכיות לוגריתמית, מאחר שהעץ ממויין לפי מספרים מזהים. על כן, מציאת הגלדיאטור שברצוננו לעדכן תעלה סיבוכיות זמן של . כעת, נעדכן את שדה Level בסיבוכיות זמן של . מאחר ועץ זה ממויין לפי מזהה (שלא השתנה), המצביע לאיבר זה מהעצים האחרים לא ישתנה גם הוא. נשמור מצביע לאיבר זה ולאחר מכן נרצה לעדכן את הגלדיאטור בשני העצים האחרים. ניגש לעץ הגלדיאטורים לפי רמה ונעשה remove לגלדיאטור הקיים. פעולה זו תקרה בסיבוכיות זמן של . כעת, נבצע פעולת insert לאותו העץ, כאשר מזהה הגלדיאטור ומזהה המאמן שלו לא השתנו אך רמתו תהיה הרמה המעודכנת (שהיא הרמה הישנה בתוספת הפרמטר levelIncrease) ובכך יכנס ישירות למיקום הרלוונטי. כמו שצויין קודם לכן, גם פעולת ה-insert תקרה בסיבוכיות זמן של . נשמור מצביע לאיבר החדש בעץ. באופן דומה, ברשותנו מצביע למאמן של הגלדיאטור ודרכו ניגש לעץ הגלדיאטורים הקטן. בנוסף, יש לנו גם מצביע לגלדיאטור עצמו בעץ הגלדיאטורים הקטן. לפיכך, נבצע גם כן פעולת remove בעץ הגלדיאטורים הקטן ב- ולאחריה פעולת insert ב- עם הרמה העדכנית (סיבוכיות זו הינה במקרה ה"גרוע" מכל כאשר כל n הגלדיאטורים נמצאים תחת מאמן זה). קיבלנו כי רמת הגלדיאטור התעדכנה בשלושת העצים הרלוונטיים וזאת בסיבוכיות זמן של . כמובן שבעת הוספת איבר והוצאת איבר מעץ נבצע גלגולים כנדרש, אך כמו שנלמד סיבוכיות הזמן של הפעולות לא תשתנה היות והגלגולים ידרשו לכל היותר כאשר h הוא גובה העץ (במקרה שלנו ) ולכן סיבוכיות הזמן תשאר זהה.

* + **StatusType GetTopGladiator(void \*DS, int trainerID, int \* gladiatorID)**

פעולה זו תתחלק לשני מקרים:

במידה ו- , נרצה להחזיר את הגלדיאטור בעל הרמה הגבוהה ביותר במערכת. לכן, ניגש לעץ הגלדיאטורים לפי רמה. היות והעץ ממויין לפי רמת הגלדיאטורים בסדר יורד, הגלדיאטור בעל הרמה הגבוהה ביותר יהיה הגלדיאטור הראשון. מאחר והמיון המשני הוא לפי המספר המזהה בסדר עולה, במידה וישנם מספר גלדיאטורים בעלי אותה רמה – נחזיר את הגלדיאטור בעל המזהה הקטן ביותר. היות ואין צורך לבצע חיפוש במקרה זה, אלא פשוט להחזיר את הגלדיאטור הראשון, פעולה זו תתבצע בסיבוכיות זמן מסדר גודל של .

במידה ו- , נרצה להחזיר את הגלדיאטור הטוב ביותר תחת המאמן הנתון לפי המזהה שקיבלנו. במקרה זה, ניגש לעץ המאמנים ונבצע חיפוש על מאמן לפי המזהה שלו. היות והעץ ממויין לפי המספרים המזהים של המאמנים, בהינתן המזהה של המאמן הרצוי נמצא אותו בסיבוכיות זמן מסדר גודל של היות וישנם k מאמנים בעץ. משמצאנו את המאמן, ניגש כעת לעץ הגלדיאטורים הקטן שלו. היות והעץ ממויין לפי רמת הגלדיאטורים בסדר יורד, הגלדיאטור בעל הרמה הגבוהה ביותר יהיה הגלדיאטור הראשון. מאחר והמיון המשני הוא לפי המספר המזהה בסדר עולה, במידה וישנם מספר גלדיאטורים בעלי אותה רמה – נחזיר את הגלדיאטור בעל המזהה הקטן ביותר. היות ואין צורך לבצע חיפוש במקרה זה, אלא פשוט להחזיר את הגלדיאטור הראשון, פעולה זו תתבצע בסיבוכיות זמן מסדר גודל של . לכן, בסך הכל מציאת המאמן והחזרת הגלדיאטור הטוב ביותר שלו יתבצעו בסיבוכיות זמן מסדר גודל של ועל כן בפרט ב-.

* + **StatusType GetAllGladiatorsByLevel(void \*DS, int trainerID, int \*\*gladiators, int \*numOfGladiator)**

פעולה זו תתחלק לשני מקרים:

במידה ו- , נרצה להחזיר את מדרג הגלדיאטורים במערכת כולה. במקרה זה, ראשית נקצה מערך בגודל n היות וישנם n גלדיאטורים במערכת כולה. לאחר מכן, ניגש לעץ הגלדיאטורים לפי רמה ונבצע סיור in-order על העץ. היות והעץ ממויין לפי הרמה בסדר יורד (ומיון משני לפי המזהה בסדר עולה) נצטרך רק להזין את כל האיברים לפי סדר הסיור. במקרה זה, סיבוכיות הזמן תהיה כריצה על כל הגלדיאטורים בעץ ועל כן תתבצע בסדר גודל של .

במידה ו- , נרצה להחזיר את מדרג הגלדיאטורים תחת מאמן יחיד לפי המזהה שקיבלנו. במקרה זה, ניגש לעץ המאמנים ונבצע חיפוש על מאמן לפי המזהה שלו. היות והעץ ממויין לפי המספרים המזהים של המאמנים, בהינתן המזהה של המאמן הרצוי נמצא אותו בסיבוכיות זמן מסדר גודל של היות וישנם k מאמנים בעץ. משמצאנו את המאמן, נקצה מערך חיצוני בגודל כמספר הגלדיאטורים תחת המאמן. לאחר מכן, ניגש לשדה המכיל את עץ הגלדיאטורים הקטן שלו ונבצע סיור in-order על העץ. היות והעץ ממויין לפי הרמה בסדר יורד (ומיון משני לפי המזהה בסדר עולה) נצטרך רק להדפיס את כל האיברים לפי סדר הסיור. במקרה זה, סיבוכיות הזמן תהיה כריצה על כל הגלדיאטורים בעץ ועל כן תתבצע בסדר גודל של .

* + **StatusType UpgradeGladiator(void \*DS, int gladiatorID, int upgradedID)**

נרצה לעדכן את המספר המזהה של הגלדיאטור. ראשית, ניגש לעץ הגלדיאטורים לפי מזהה. היות והעץ ממויין לפי מזהה, בהינתן המזהה של הגלדיאטור נוכל למצוא אותו בסיבוכיות זמן של . כעת, ניגש למצביע אל הגלדיאטור בעץ הגלדיאטורים לפי רמות ונעדכן שם את המזהה שלו למזהה החדש שקיבלנו. היות ועץ זה ממויין לפי רמה, מיקומו של הגלדיאטור לא ישתנה. בדומה לכך, ניגש גם למצביע אל הגלדיאטור בעץ הגלדיאטורים הקטן ונעדכן גם שם את רמתו לרמה החדשה. גם עץ זה ממויין לפי רמות ולכן גם מיקומו בעץ הזה לא ישתנה. מאחר וניגשנו ישירות למצביע בשני עצים אלה, הפעולות התבצעו בסיבוכיות זמן של . לבסוף, נשאר לנו רק לעדכן את המזהה שלו בעץ הגלדיאטורים לפי מזהה, שם כבר מצאנו את מיקומו ה"ישן". נשמור את כל פרטי הגלדיאטור ונבצע remove מהעץ. כידוע לנו, פעולת remove תתבצע בסיבוכיות זמן של . לאחר מכן, כאשר ברשותנו כל פרטי הגלדיאטור המעודכנים (רק המזהה השתנה), נבצע insert לעץ עם הפרטים העדכניים. כידוע, פעולת insert תתבצע בסיבוכיות זמן של ותמקם את הגלדיאטור בצורה ממויינת לפי המזהה החדש. נשים לב כי לאחר מיקומו בעץ אין צורך לשנות את המצביעים מאחר ומיקומו בעצים האחרים לא השתנה. כמובן שבעת הוספת איבר והוצאת איבר מעץ נבצע גלגולים כנדרש, אך כמו שנלמד סיבוכיות הזמן של הפעולות לא תשתנה היות והגלגולים ידרשו לכל היותר כאשר h הוא גובה העץ (במקרה שלנו ) ולכן סיבוכיות הזמן תשאר זהה. לכן, בסך הכל הפעולה תתבצע בסיבוכיות זמן מסדר גודל של .

* + **StatusType UpdateLevels(void \*DS, int stimulantCode, int stimulantFactor)**

??

* + **void Quit(void \*\*DS)**

נרצה לשחרר את כל המבנה. ראשית, נשחרר את עץ הגלדיאטורים לפי רמה. נעבור בסיור post-order על העץ ונשחרר (delete ולא remove, מאחר ואין ברצוננו לשמור על מיון העץ) כל איבר לפי סדר הסיור. נבצע זאת בסיור post-order על מנת להמנע משחרור איבר לפני שחרור בניו. מאחר וגודל העץ הוא n, פעולת שחרור זו תתבצע בסיבוכיות זמן מסדר גודל של . באופן דומה ובסיבוכיות זמן זהה, נעבור גם על עץ הגלדיאטורים לפי מזהה. לאחר מכן, נעבור בסיור post-order על עץ המאמנים. לפי סדר הסיור, בכל איבר נבצע שחרור לעץ הגלדיאטורים הקטן שלו. גם את עץ הגלדיאטורים הקטן נשחרר באופן זהה לדרך שהסברנו קודם לכן. לכן, שחרור עץ גלדיאטורים קטן יתבצע בסיבוכיות זמן מסדר גודל של ועל כן שחרור כל עצי הגלדיאטורים הקטנים יתבצעו בסיבוכיות זמן מסדר גודל של . לאחר שסיימנו לשחרר את עץ הגלדיאטורים הקטן של המאמן, נשחרר את המאמן עצמו ונעבור למאמן הבא לפי סיור post-order. מאחר וישנם k מאמנים, פעולת שחרור כל מאמן תתבצע בסיבוכיות זמן של לפי שחרור עץ הגלדיאטורים הקטן שלו. לכן, שחרור סך כל המאמנים ירוץ על k מאמנים ובכל אחד מהם על גלדיאטורים שיסכמו בסה"כ ל-n גלדיאטורים. לפיכך, פעולת שחרור המאמנים ועצי הגלדיאטורים הקטנים תתבצע בסדר גודל של . כתוצאה מכך, פעולת Quit כולה תתבצע בסיבוכיות זמן של .