תרגיל רטוב 1: מגישים: אסף לובטון 209844414 ועדן דמבינסקי 212227888.

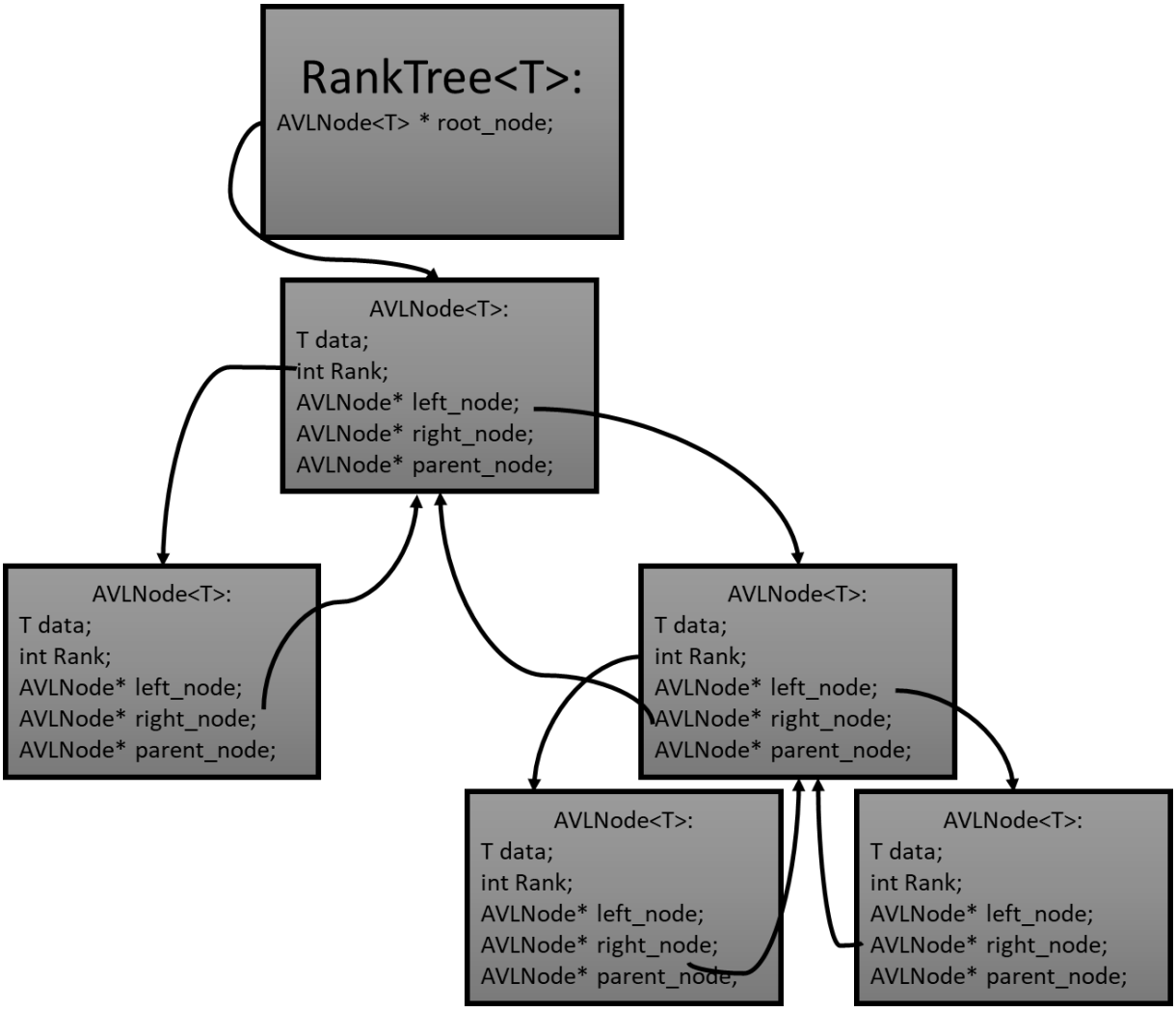
עבור תרגיל זה נחזיק טבלת ערבול (מסוג **chain hashing**) ועץ דרגות, כאשר את עץ הדרגות נממש באמצעות עץ החיפוש (מסוג **AVL**) הגנרי שמימשנו בתרגיל הבית הקודם. עבור טבלת הערבול מימשנו שרשרת חוליות גנרית (כפולה-כל חוליה מצביעה לחוליה ממנה הגיעה).

את עץ הדרגות מימשנו באופן הבא:

לחוליה הגנרית בעץ החיפוש, יש מצביע לחוליה ימנית, שמאלית, אבא, שדה המכיל מצביע לטיפוס הגנרי שיכיל את התוכן, שדה דרגה אשר פירוש מספר הצמתים בתת העץ של צומת זה.

לעץ הפעולות הסטנדרטיות שראינו בהרצאה ובתרגולים: המימוש תואם לאופן שבו למדנו ולכן לא נסביר את אלגוריתמי הכנסה הוצאה וחיפוש שכן מדובר בחומר שנלמד. הכנסה (**insert** מבוססת על העמסת אופרטור =<,== עבור הטיפוס הגנרי), חיפוש (**find** מבוסס על העמסת אופרטור = עבור הטיפוס הגנרי), מחיקה (**delete** מבוסס על אופרטור >,<), גלגול לימין, גלגול לשמאל, חישוב גובה, חישוב פקטור איזון. רק נציין כי הוספת שדה הדרגה לכל צומת לא משנה את סדר הסיבוכיות של כל פעולה.

הכנסה חיפוש ומחיקה כל אלה בסיבוכיות זמן לוגוריתמית למספר הצמתים בעץ(שווה לגובה העץ) כפי שנלמד והוכח בכיתה.



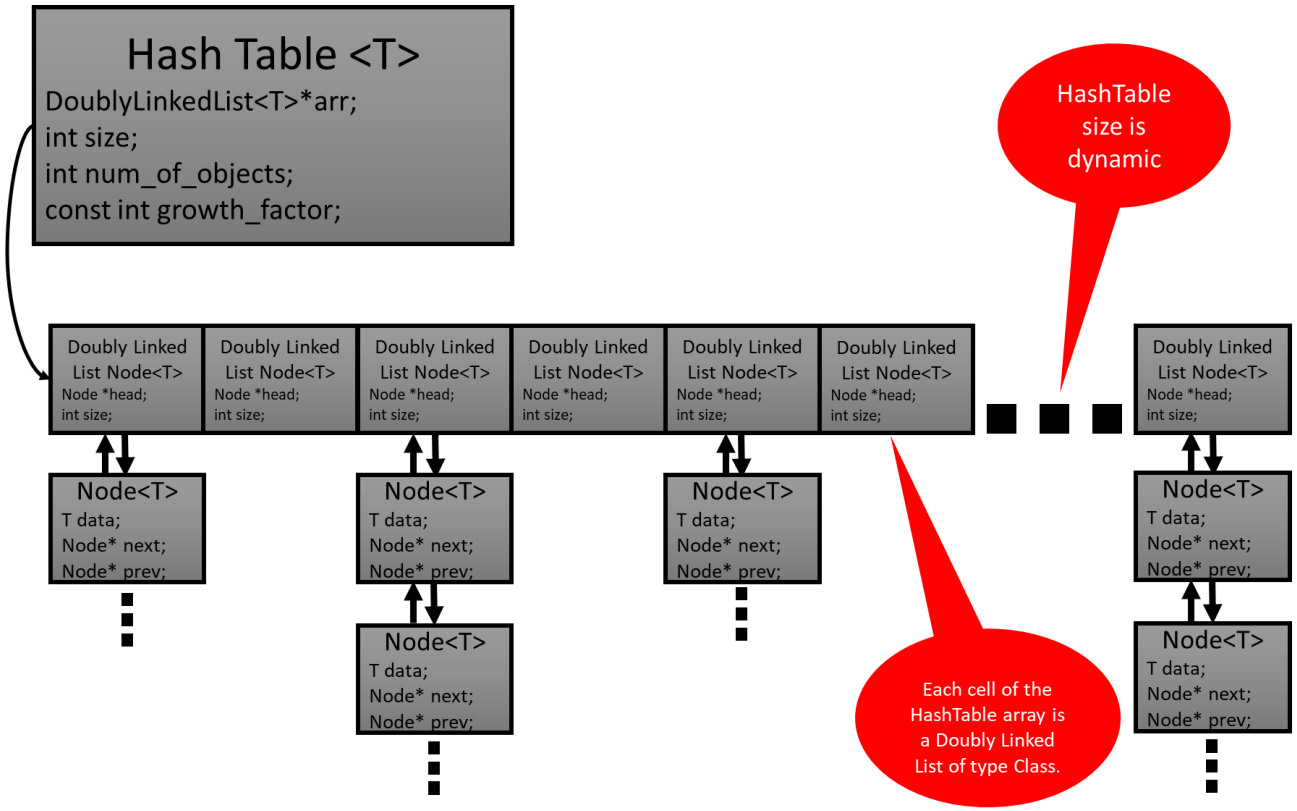
את טבלת הערבול הגנרית מימשנו באופן הבא:

יש לנו מערך שגודלו משתנה באופן דינאמי, גודל המערך גדל פי פקטור ההגדלה אם מספר האיברים שיש בטבלה חלקי גודלה (פקטור העומס) גדול מפקטור ההגדלה אז אנחנו מגדילים את הטבלה פי פקטור ההגדלה. כלומר אם מילאנו את הטבלה ב100 איברים כאשר גודל המערך הוא 10 נגדיל את המערך להיות מגודל 1000. באופן דומה אם גודל המערך חלקי מספר האיברים שיש בטבלה גדול מפקטור ההגדלה נקטין את המערך לפי פקטור ההגדלה. וקטן פי 10 אם מילאנו מאית ממנו. יתרה מכך כל תא במערך הדינאמי הוא מסוג שרשרת חוליות כפולה מסוג הטיפוס של הטבלה כלומר אם בטבלה יש 10 איברים וגודל המערך כעת הוא 100 נקטין את גודל המערך להיות 10.

פירוט על שרשרת החוליות ופעולותיה:

הכנסה של איבר לשרשרת מתבצעת בראש הרשימה לכן היא מסדר גודל **O(1)** . מחיקה של איבר היא מסדר גודל לינארי למספר האיברים ברשימה מאחר ורצים על הרשימה ומסירים את האיבר לכן הסיבוכיות מסדר גודל **O(n)**. בנוסף יש לנו שדה של מספר האיברים ברשימה שעלינו לעדכן אותו בכל הוספה או הסרה של איבר אך תמיכה בפעולה זו מוסיפה לנו מספר קבוע של פעולות לכל פעולת הוספה או הסרה ולכן לא משנה את הסיבוכיות שלהן.

את פעולות ההוספה , הסרה, הגדלה, הקטנה של טבלת הערבול מימשנו בהתאם לנלמד בכיתה ולכן לא נפרט את האלגוריתמיקה מאחורי מבנה הנתונים רק נזכיר שהסיבוכיות של כל הפעולות יחדיו היא **O(1)** משוערך במידע זה נשתמש בהוכחות הסיבוכיות בהמשך ההסבר.



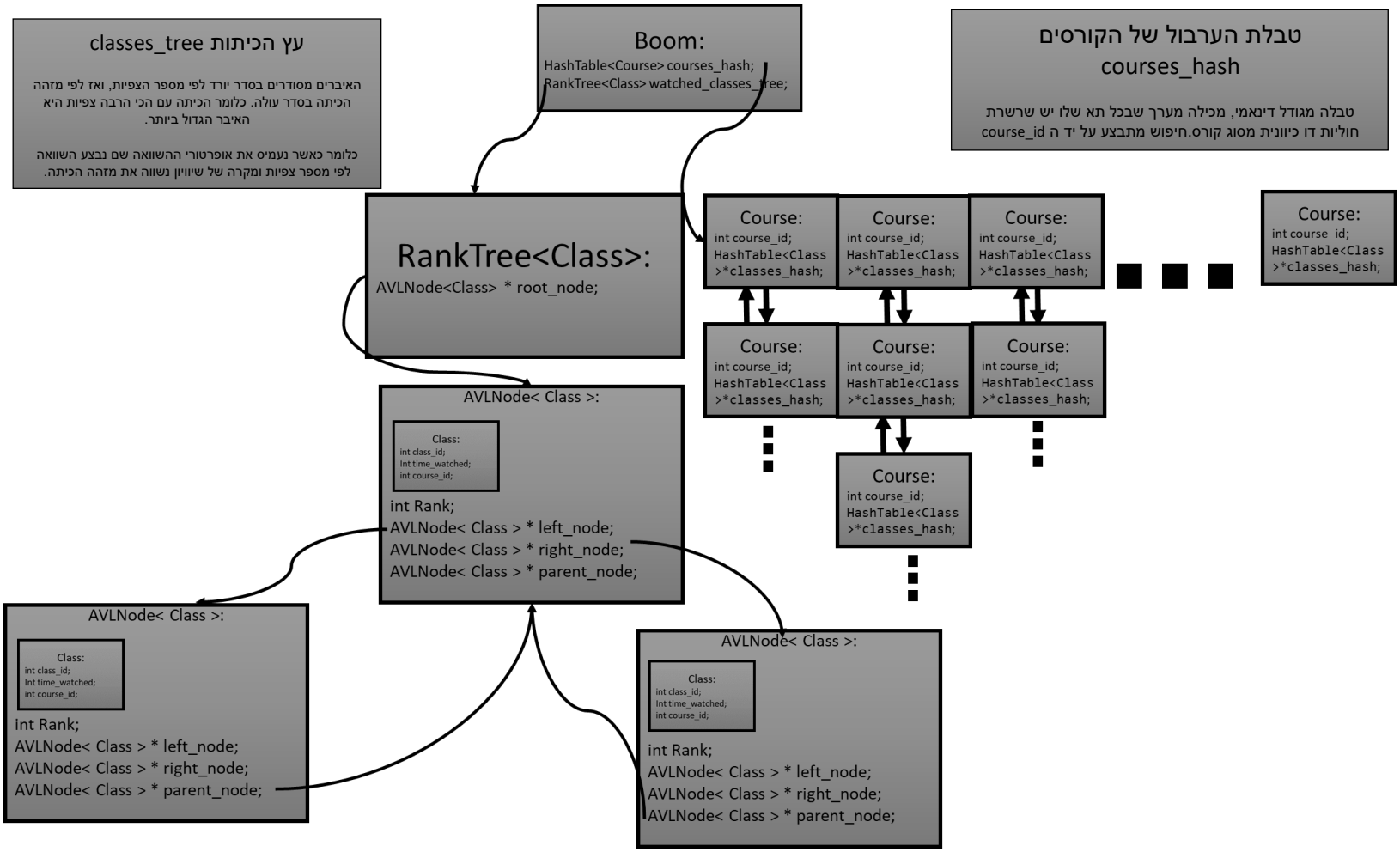
כעת לאחר שהסברנו על מבני הנתונים שבנינו עבור התרגיל נפרט על הטיפוסים שבנינו וכיצד שילבנו אותם עם מבני הנתונים על מנת לעמוד בדרישות התרגיל.

בנינו טיפוס מסוג קורס (**Course**): עבור כל קורס שמרנו מספר שיהיה מזהה הקורס וטבלת ערבול של הכיתות בקורס זה.

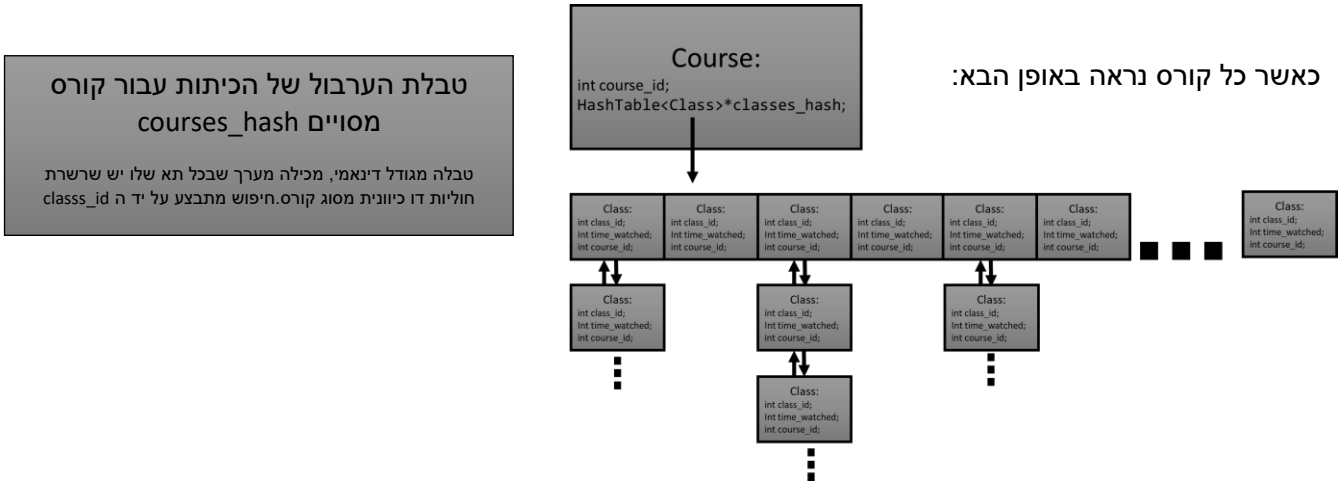
טיפוס מסוג כיתה (**Class**): עבור כל כיתה שמרנו מספר שיהיה מזהה כיתה, כמות הזמן הנצפה, ואת מזהה הקרוס אליו היא שייכת.

טיפוס (**Boom**): מחזיק טבלת ערבול מסוג **Course** ועץ חיפוש מסוג **Class** (העץ ממויין לפי זמני הצפייה ובמקרה של שיוויון לפי לפי מזהה השיעור) כאשר רק כיתה שמספר הצפיות בה חיובי מתווספת לעץ זה.

עבור מבנה הנתונים **DS** נבנה מופע אחד של **Boom**. ניתן לראות איור של שני העצים בעמוד הבא.



כאשר כל קורס נראה באופן הבא:



הסבר על הפעולות אותן נדרשנו לממש: **n** מספר הקורסים במערכת, **m** מספר ההרצאות של קורס מסויים, **M** מספר ההרצאות במערכת.

void * Init():

הפעולה מחזירה מצביע לטיפוס מסוג **Boom**, הפעולה קוראת לבנאי שלו אשר קורא לבנאי של טבלת הערבול אשר בונה מערך בגודל של 10 תאים, מאחר וגודל המערך הוא מסדר גודל קבוע, כל תא במערך מאותחל להיות **null** לכן יצירת טבלת הערבול היא מסדר קבוע של פעולות. בנוסף קוראת לבנאי של עץ הדרגות. כאשר הבנאי של העץ הגנרי שם ערך **null** בשורש העץ על מנת לאתחל אותו. לכן מדובר בסהכ במספר קבוע של פעולות לכן סיבוכיות הזמן היא מסדר גודל **O(1)**.

StatusType AddCourse(void* DS, int courseID):

על מנת להוסיף מופע של הטיפוס קורס מדובר במספר פעולות קבוע על מנת ליצור את המופע שלו, כעת נותר להוסיף את הקורס לטבלת הערבול של הכיתות וכפי שפירטנו למעלה הוספה לטבלת הערבול מתבצעת ב**O(1)** משוערך. יתרה מכך יצירת מופע של קורס קורס לוקחת מספר קבוע של פעולות ולכן הסיבוכיות הדרושה לכך **O(1)**.

לכן ראינו כי הוספה של קורס חדש היא בסיבוכיות **O(1)** בממוצע על הקלט, משוערך.

StatusType RemoveCourse(void *DS, int courseID):

על מנת למחוק קורס עלינו למחוק את הקורס מטבלת הערבול של הקורסים, וגם לדאוג למחיקה של כל כיתה שבקורס מעץ הכיתות.

תחילה נחפש את הקורס בטבלת הקורסים, חיפוש כזה כפי שלמדנו נעשה ב**O(1)** משוערך. כעת כשמצאנו את הקורס שברצוננו למחוק קודם כל נמצא את כל הכיתות השייכים לקורס זה ונמחק אותם מעץ הכיתות. נזכיר שהוספה של כיתה לעץ מתבצעת רק אם היו לו צפיות, וכי האינדקס (**class_id**) המתאר את הכיתה ניתן לה לפי מספר הכיתות בקרוס בעת הוספתה לקורס. מאחר ואין אפשרות למחוק כיתה מקורס קבוצת ה**class_id** המתארת את הכיתות שנמצאות בקרוס היא מ0 ועד מספר הכיתות שיש בקורס. עובדה זו עוזרת לנו מאחר ועלינו לחפש את כל הכיתות שהוספנו לעץ הכיתות ולמחוק אותן מהעץ, את החיפוש בעץ אנו עושים לפי ה**class_id** ואם מספר הצפיות של אותו הקורס .

המחיקה של כל כיתה מעץ הכיתות היא לוגוריתמית למספר הכיתות בעץ (ל**M**), לכן עבור כל כיתה נבצע **log(M)** סה"כ **O(mlog(M))** פעולות על מנת למחוק את כל הכיתות מעץ הכיתות.

מחיקת טבלת הערבול כלומר את מערך הכיתות של הקורס תיקח סדר גודל של סיבוכיות **O(m)**.

לכן סה"כ נקבל כי מציאת הקורס שעלינו למחוק ב**O(1)** משוערך, מחיקת כל **M** הכיתות מעץ הכיתות ב**O(mlog(M))** ומחיקת טבלת הערבול ב**O(m)** כלומר נקבל **O(mlog(M))** משוערך כמבוקש.

StatusType AddClass(void* DS, int courseID, int* classID):

על מנת להוסיף כיתה עלינו למצוא את הקורס שאליו אנו רוצים להוסיף אותה בטבלת הערבול של הקורסים ניתן לעשות זאת ב **O(1)** משוערך, לאחר מכן יש להוסיף את הכיתה לטבלת הכיתות של הקורס, המזהה של הכיתה נקבע על פי מספר הכיתות שיש בקורס. ניתן לקבל שדה זה באמצעות שדה num_of_objects של טבלת הערבול של הקורס. ראינו כי הוספה לטבלת ערבול נעשית ב **O(1)** משוערך, לכן סה"כ קיבלנו שהסיבוכיות היא **O(1)** משוערך .

StatusType WatchClass(void *DS, int courseID, int classID, int time):

מאחר ואנחנו לא מוסיפים כיתה לעץ הכיתות אם מספר הצפיות של הכיתה עומד על אפס. אם לא צפו בה עדיין נוסיף חוליה חדשה של כיתה זו לעץ הכיתות עם הזמן שהתקבל עבורה, מאחר ומסופק לנו המזהה קורס והמזהה כיתה אנו נבצע חיפוש בעץ הכיתות ונוסיף אותה במקום המתאים לה על פי זמן הצפייה שלה (החיפוש מתבצע באמצעות האופרטור = ו> שאותם העמסנו בשתי המחלקות של כיתה ושל קורס). את ההוספה הזו אנחנו מבצעים ב**O(log(M))** כפי שלמדנו. אם כבר צפו בכיתה זו כלומר מספר הצפיות בה גדול מאפס היא בהכרח נמצאת בעץ לען עלינו להוסיף את זמן הצפייה שהתקבל עבורה לערך הקיים בעץ הכיתות למצוא כיתה בעץ הכיתות לוקח סדר פעולות לוגוריתמי למספר הכיתות במערכת כלומר **O(log(M))** פעולות. העדכון עצמו לוקח מספר קבוע של פעולות. לאחר שמצאנו את הכיתה אנחנו שומרים העתק שלה עם זמן הצפייה החדש, מוחקים את הכיתה מהעץ הכיתות ומוספים אותה מחדש לעץ עם הערכים העדכניים, מאחר וחיפוש מחיקה והכנסה כולם מתבצעים בסדר גודל לוגוריתמי למספר הכיתות בעץ הכיתות סך הכל לקח לנו **O(log(M))**.

פעולה נוספת שנעשה היא לעדכן את הדרגות של הצמתים לאורך ההוספה והמחיקה של איבר מהעץ, כל עדכון כזה לוקח מספר קבוע של פעולות לאורך מסלול החיפוש ולכן משאיר את פעולות ההכנסה והוצאה בסדר גודל **O(log(M))** פעולות.

כעת עלינו לעדכן את הכיתה בטבלת הכיתות עבור הקורס ששמור בטבלת הקורסים עדכון זה נעשה ב **O(1)** משוערך מאחר ומדובר בפעמיים פעולת חיפוש בטבלת ערבול ואז בעדכון שדה אשר נעשה במספר קבוע של פעולות פעם אחת ולכן לא משפיע על הסיבוכיות. נדגיש כי החיפוש בטבלת הקורסים נעשה בסיבוכיות **O(1)** משוערך ביחס למספר הקורסים שקטן ממש ממספר הכיתות במערכת לכן סה"כ קיבלנו שהפעולה כולה לוקחת **O(log(M))** משוערך.

במקרה בו אין בכלל כיתות במערכת הסיבוכיות הופכת להיות **O(1)** משוערך מאחר ולא מתבצע חיפוש בעץ הכיתות. לסיכום נקבל **O(log(M+2))** בממוצע על הקלט.

StatusType TimeViewed(void *DS, int courseID, int classID, int *timeViewed):

על מנת לקבל את משך הצפייה של של כיתה מסויימת, נחפש בטבלת הערבול לפי מזהה הקורס ניתן לעשות זאת ב **O(1)** משוערך, לאחר מכן יש לחפש את הכיתה בטבלת הערבול של הכיתות של הקורס שמצאנו לפי מזהה הכיתה פעולה זו נעשית גם היא ב **O(1)** משוערך. לכן סהכ שמרנו על **O(1)** משוערך.

StatusType GetIthWatchedClass(void* DS, int i, int* courseID, int* classID):

על מנת לבצע חלק זה ביצענו את ההוספה של תכונת הדרגות לעץ החיפוש. תכונה זו מאפשר לנו להביא את האיבר בעל האינדקס ה**i** ב **log(M)** כפי שלמדנו בתרגול. מימשנו את פונקציית **select** כפי שהוצגה לנו. לכן על מנת להביא את הכיתה המובקשת יעלה לנו **O(log(M+2))** במקרה הגרוע, כאשר **M** הוא מספר השיעורים במערכת בזמן ביצוע הפעולה. נציין באופן שאנחנו מימשנו הצלחנו להוריד את הסיבוכיות אפילו יותר בכך שבעץ הכיתות שלנו מופיעות רק כיתות בעלות מספר חיובי של צפיות.

<u>סיבוכיות מקום:</u>

אנחנו מחזיקים עץ חיפוש אחד, עץ הכיתות הוא בגודל הכיתות שיש להן יותר מ0 צפיות, לכן גודלו חסום על ידי **m** שזה מספר כלל הכיתות במערכת. גודל הטיפוס בעץ זה הוא 3 **int**, ואנו שומרים מצביע לטיפוס ו3 מצביעים עבור בן שמאלי בן ימני ואב, לכן סהכ סיבוכיות המקום עבור עץ הכיתות היא לכל היותר **O(m)**.

עבור טבלת הערבול אנחנו מחזיקים לכל היותר מערך בגודל פקטור הגדילה כפול מספר הקורסים במערכת. ולכל קורס אנחנו מחזיקים לכל היותר מערך בגודל פקטור הגדילה כפול מספר הכיתות בקורס. כלומר לכל היותר אנחנו מחזיקים את פקטור הגדילה בריבוע כפול סכום הכיתות בכל קרוס כלומר כפול כל הכיתות במערכת. נזכור כי פקטור הגדילה הוא קבוע לכן לכל היותר סיבוכיות המקום עבור טבלת הערבול היא מגודל **O((growth_factor^2)*m .** כלומר **O(m)** כל איבר בטבלת הערבול הוא מסדר גודל קבוע של משתנים ולכן מכפיל בקבוע את סיבוכיות המקום ולא משנה אותה.

במקרה שבו טרם הוספו למנה כיתות בעלות צפיות נקבל שגודל טבלת הערבול של הקורסים חסום על ידי **O((growth_factor)*n)** כלומר **O(n)**.

לכן סה"כ מדובר בסיבוכיות מקום **O(n+m)**.

<u>void Quit(void **DS):</u>

עבור פעולה זו עלינו למחוק את כל מבני הנתונים, מאחר וראינו שסיבוכיות המקום היא **O(n+m)** נסיק כי על מנת למחוק את כל מבני הנתונים סיבוכיות הזמן תהיינה **O(n+m)**.