**תיאור המבנה:**

המבנה שלנו בנוי מטבלת האש ועץ AVL : טבלת ההאש מכילה את הקורסים, עץ של השיעורים הנצפים

טבלת ההאש של הקורסים: זוהי טבלת האש שממושת בעזרת רשימות מקושרות, שהמפתח שלה הוא מספר הקורס, ופונקציית ההאש שלה היא מודולו עם גודל המערך, הטבלה מכילה עבור כל מספר קורס את המערך שלו

"מערך הקורס": מערך דינאמי שמכיל בתוכו את מספר הצפיות לכל שיעור, ובנוסף מכיל נתונים על הקורס כמו מספר הקורס או מספר השיעורים בקורס

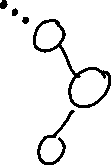
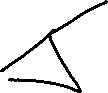
ClassTuple: מבנה שמכיל בתוכו שלושה משתנים, זמן הצפיה של אותו שיעור, מספר השיעור ומספר הקורס שאותו שיעור שייך אליו. מוגדרות אליו פעולות גדול, קטן ושווה לפי ההגדרה של איזה שיעור צריך לשים קודם במערך השיעורים הכי נצפים(יותר גדול מי שיש לו זמן צפיה יותר גדול, במקרה של שיווין, זה בעל מספר הקורס הקטן יותר, ובמקרה של שיווין שני זה בעל מספר השיעור הקטן יותר)

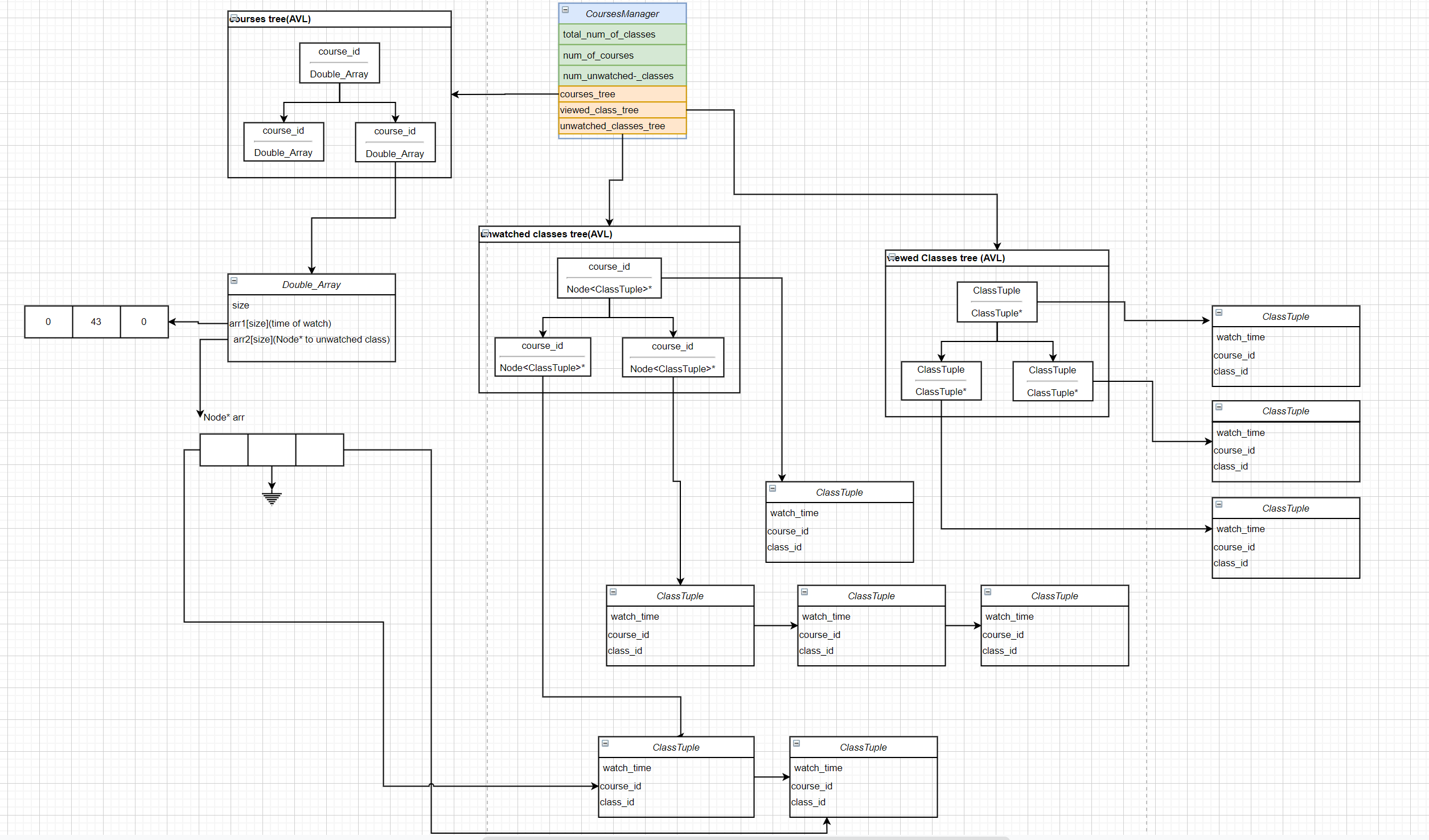
עץ השיעורים הנצפים: זהו עץ AVL מדורג שהאיברים הם ClassTupleים כאשר בהכרח זמן הצפיה של כל שיעור בו גדול ממש מ0

מימוש עץ הAVL:

מימוש העץ AVL עם דרגות, הוא עם שימוש באלגוריתמים בראינו בהרצאה, אם מעט שינוים.

* הוספת/הוצאת איבר: לאחר הוספת/הוצאת איבר לפי האלגוריתם הנלמד בכיתה, "נרד" במורד העץ עד לאיבר הכי גדול ועד לאיבר הכי קטן ונשמור מצביע אליהם. גובה עץ AVL הוא לכל היותר לוג מספר האיברים בו ולכן עדכון ושימרת מצביעים אלו לאחר הוספת/הוצאת איבר אינה פוגעת בסיבוכיות הזמן של פעולות אלו שכן הן בסיביוכיות זמן של לוג מספר האיברים
* הוספת פונקציות סיור inorder החל מצומת מסוימת בעץ (שימוש להוצאת השיעורים הכי נצפים): הפונקציה מקבלת את מספר האיברים שתרצה מהעץ ומצביע לצומת כלשהי בעץ (אנו ניתן לה את הצומת הכי גדולה או הכי קטנה) הפונקציה עושה סיור inorder כפי שראינו בכיתה החל מהצומת שניתנה לה ועוצרת במידה ועברה במספר האיברים המבוקש. אם אחרי סיור בכלל תת העץ שניתן לה לא עברנו בכמות הצמתים הדרושה היא "עולה" לאב של הצומת הנתונה ועושה את סיור הinorder בבן השני של אותו אב. אם גם זה לא מספיק היא "עולה" לאב של האב ופועלת באותו עיקרון בצורה רקורסיבית. היא תעבור במהלך הסיור רק בכמות האיברים הנדרשת ולכן סיבוכיות הזמן שלה היא כזו של כמות האיברים המבוקשת.



תרשים סכמתי של המבנה במצב ממוצע: **לשנות**

גם לשני אלו יש מצביע ל"מערך כפול" אבל לא משורטט בשביל לשמור על תרשים ברור יותר

* + - * ע
* **עע**

**Init()**

האלגוריתם:

1. נאתחל את מבנה הנתונים ריק- כלומר טבלה ריקה, ועץ ריק, ואת המשתנים נאתחל ל0
2. נחזיר מצביע במקרה של הצלחה וnull במקרה של כישלון

סיבוכיות זמן האלגוריתם:

1. O(1)- מספר סופי וקבוע של פעולות
2. O(1)- מספר סופי וקבוע של פעולות

לכן **בסה"כ** סיבוכיות זמן האלגוריתם O(1)

סיבוכיות מקום האלגוריתם תהיה O(1) מכיוון שאנחנו מקצים מערך בגודל קבוע וחוץ ממנו מספר קבוע נוסף של משתנים.

**StatusType AddCourse (void \*DS, int courseID, int numOfClasses)**

האלגוריתם:

1. נוודא את תקינות הקלט ואם לא תקין נחזיר ערך מתאים
2. נוודא שהקורס אינו קיים במבנה שלנו על ידי בדיקה אם הקורס נמצא בטבלה
3. נוסיף לטבלה במקום מספר הקורס מערך חדש עבור הקורס.

* הסבר על מבנה "המערך שיעורים" שיצרנו:

ביצירת המצביע אנו יוצרים שלוש מערכים**?**

1. עדכון משתנים נוספים של המבנה כגון מספר השיעורים הכולל, מספר השיעורים הלא נצפים.

סיבוכיות זמן האלגוריתם:

1. בדיקה של משתנים o(1)
2. בדיקת ערך קיים בטבלה, ראינו בהרצאה o(log(n)) כאשר n-מספר הקורסים הקיימים כבר במבנה
3. הוספת ערך לטבלה, כנ"ל o(log(n)) יצירת "המערך שיעורים" o(m) כאשר m=מספר השיעורים שיש בקורס, משום שאנחנו מקצים ומאתחלים מערך בגודל m ויוצרים רשימה בגודל m .
4. עדכון של מספר משתנים סופי o(1)

* **סה"כ:** o(m+log(n)) כאשר n=מספר הקורסים הקיימים במבנה, m=מספר השיעורים שיש לקורס המוסף.

סיבוכיות מקום האלגוריתם:

בהוספת ערך לטבלה אנחנו במקרה הגרוע יוצרים רשימה חדשה-O(1) ואז יוצרים שלושה מערכים ומספר קבוע של מערכים-O(m) בנוסף תהיה פעולה רקורסיבית ברשימות בטבלה שמקרה הכי גרוע O(N) ולכן **סה"כ:** O(m+N)

**StatusType RemoveCourse(void \*DS, int courseID)**

האלגוריתם:

1. נוודא את תקינות הקלט ואם הוא לא תקין נחזיר ערך מתאים
2. נמצא את הקורס בטבלה אם הוא לא נמצא נחזיר ערך מתאים
3. במידה ומצאנו מקודם את הקורס, ניגש למערך השיעורים של הקורס, עבור כל שיעור שנצפה, נשחרר אותו מעץ השיעורים הנצפים.
4. נשחרר את המערכים של הקורס.
5. נחזיר הצלחה.

סיבוכיות זמן האלגוריתם:

1. O(1) -בדיקת מספר משתנים סופי
2. O(1) בממוצע -חיפוש בטבלת האש
3. O(m\*log(M)) כאשר m מספר ההרצאות של הקורס וM מספר ההרצאות הכולל במערכת- המקרה הכי גרוע יהיה כאשר כל אחד מהשיעורים שבמנה יהיה בעל זמן צפיה חיובי ממש כלשהו. ולכן לכל שיעור בקורס שנרצה להוציא נאלץ להוציא את האיבר שלו מעץ השיעורים הנצפים וזה יהיה בסיבוכיות של O(logM) מכיוון שזוהי הוצאה מעץ AVL שבמקרה הכי גרוע יש לו M איברים.
4. O(m)- שחרור של 3 מערכים בגודל m.
5. O(1)- פעולה אחת

* **בסה"כ**: הסיבוכיות זמן תהיה בסדר גודל של O(mlogM) כנדרש

סיבוכיות מקום של האלגוריתם:

לא נדרשת הקצאה של מקום או פעולה רקוסיבית פרט לפעולות רקורסיביות של עץ או רשימה כמו של הוצאה ובמקרה הכי גרוע זו תיהיה הוצאה מעץ השיעורים הנצפים כאשר כל השיעורים במבנה נצפו כבר. לכן סיבוכיות מקום: O(log(M)) כאשר M=מספר השיעורים הכולל במבנה והוצאה מרשימה שבמקרה הכי גרוע O(N) לכן בסה"כ O(logM+N)

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

האלגוריתם:

1. נוודא את תקינות הקלט ואם לא תקין נחזיר ערך מתאים
2. נוודא שהקורס קיים במבנה. אם לא נחזיר ערך מתאים
3. נוודא שלקורס הנ"ל יש מספר שיעור כזה על ידי בדיקת כמות מספר הכיתות. אם אין נחזיר ערך מתאים
4. במידה וצפו בו לפני כן, נוציא את הצומת המתאימה לשיעור מעץ השיעורים הנצפים
5. נעלה את זמן הצפייה של השיעור והקורס
6. נכניס לעץ הכיתות הנצפות את הכיתה הזאת לאחר עדכון זמן הצפייה בה.
7. נחזיר הצלחה (במידה ולא הייתה שגיאת הקצאה)

סיבכיות זמן האלגוריתם:

1. O(1) – בדיקה של מספר סופי של משתנים
2. O(1) בממוצע - חיפוש הקורס בטבלה. כאשר n=מספר הקורסים הקיימים כבר במבנה.
3. O(1)- חיפוש הקורס בטבלה ובדיקה של גודל המערכים שלו בעזרת בדיקת משתנה הגודל של מבנה "המערך הכפול"
4. O(log(M))- בסיכוי הגרוע ביותר כל השיעורים במבנה נמצאים כבר בעץ השיעורים הנצפים ולכן נצטרך להוציא איבר מעץ AVL בגודל M. כאשר M= מספר השיעורים הכולל שנמצא במבנה.
5. O(1)-מספר קבוע של פעולות
6. O(log(M)) בסיכוי הגרוע ביותר כל השיעורים במבנה נמצאים כבר בעץ השיעורים הנצפים ולכן נצטרך להכניס איבר בעץ AVL בגודל M. כאשר M= מספר השיעורים הכולל שנמצא במבנה.

**סה"כ** סיבוכיות זמן של O(log(M)) במקרה הגרוע (M>>n) כאשר M זה מספר השיעורים הכולל במערכת

סיבוכיות מקום האלגוריתם:

פה ההקצאה היחדיה היא של צומת בודדת בעץ השיעורים הנצפים ואין פעולות רקורסיביות פרט לאלו של פעולות העץ והרשימה. העץ הכי גדול במקרה הגרוע ביותר יהיה עץ השיעורים הנצפים עם M איברים לכן סיבוכיות מקום של רקורסיה בו היא O(log(M)) הרשימה מקרה הכי גרוע בגודל O(N) ולכן סה"כ O(log(M)+N)

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

האלגוריתם:

1. נבדוק את תקינות הפלט- אם הוא לא תקין נחזיר את הערך המתאים
2. נמצא את הקורס בטבלת ההאש- אם הוא לא נמצא נחזיר את הערך המתאים
3. נוציא מ"המערך שיעורים" של הקורס את זמן הצפיה השמור במקום התואם למספר השיעור במערך הראשון
4. נחזיר הצלחה(במידה ולא הייתה שגיאת הקצאה)

סיבוכיות זמן האלגוריתם:

1. O(1)- מספר סופי קבוע של פעולות יחידות
2. O(1) בממוצע -חיפוש בטבלת האש
3. O(1)- חיפוש בצומת בעץ(AVL) הקורסים וגישה למערך בO(1)
4. O(1)- מספר סופי קבוע של פעולות יחידות

* **סה"כ** הסיבוכיות זמן תהיה O(1) בממוצע על הקלט כנדרש.

סיבוכיות מקום:

תיתכן הפעלה של פעולה רקורסיבית באחת הרשימות בטבלת ההאש במקרה הכי גרוע בO(N)

**StatusType GetIthViewedClasses(void \*DS, int numOfClasses, int \*courses, int \*classes)**

האלגוריתם:

1. נבדוק את תקינות הקלט ונחזיר ערך מתאים
2. נבדוק אם קיימים במבנה שלנו מספיק שיעורים על ידי בדיקת המשתנה המתאים. אם לא קיימים נחזיר ערך מתאים.
3. נחפש את האיבר הi בגודלו בעץ המדורג של הקורסים שנצפו כבר
4. נחזיר הצלחה (במידה ולא הייתה בעיית הקצאה)

סיבוכיות הזמן של האלגוריתם:

1. O(1) – פעולה אחת של בדיקת מצביע
2. O(1) – בדיקת משתנה יחיד
3. O(logM) – במקרה הכי גרוע כל השיעורים נצפו ולכן אנחנו מחפשים את האיבר הi בגודלו בעץ מדורג בעל (M=מספר השיעורים הכולל) איברים.
4. O(1) – פעולה אחת

**סה"כ:** O(log(M+2)) כאשר M זה מספר השיעורים הכולל

סיבוכיות מקום של האלגוריתם:

אין הקצאות והפעולה הרקורסיבית של פונקצית המציאה של הצומת הi בעץ היא לכל היותר לוג(מספר השיעורים המבוקשים) במידה וכולם נמצאים בעץ השיעורים הנצפים, במקרה הכי נורא הצטרך לעבור על כל הגובה של העץ כדי למצו את האיבר הi בגודלו(כי זה עץ דרגות)

לכן:O(log(M))

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

האלגוריתם:

1. נבדוק את תקינות הקלט ונחזיר ערך מתאים
2. על כל צומת בעץ הקורסים אנו משחררים את "המערך שיעורים"
3. משחררים את טבלת ההאש על ידי שיחרור של כל הרשימות
4. משחררים את כל הצמתים בעץ השיעורים הנצפים

סיבוכיות הזמן של האלגוריתם:

1. O(1)- בדיקת מצביע אחד
2. O(M)- שיחרור כל המערכים שגודלם הכולל M- סך כל השיעורים במבנה.
3. O(N )- מחיקת לכל היותר N-מספר הקורסים הכולל איברים מרשימות בטבלת ההאש
4. O(M)- מחיקת כל השיעורים במקרה הכי גרוע שכל השיעורים נצפו

* **סה"כ:** O(M+N) כאשר M=מספר השיעורים במבנה וN=מספר בקורסים במבנה.

סיבוכיות מקום של האלגוריתם:

אין הקצאה אלא רק פעולה רקורסיבית של מחיקת איברים מהעץ שעומקה היא לכל היותר גובה העץ. המקרה הכי גרוע העץ הכי גדול שיכול להיות הוא עץ השיעורים הנצפים בעל M(מספר השיעורים הכולל במערך) ולכן גובהו לכל היותר log(M).

לכן סיבוכיות מקום:O(logM) כאשר M-מספר השיעורים במבנה.

**סיבוכיות מקום כוללת:**

גודל טבלת ההאש הוא בסה"כ N-מספר הקורסים הכולל

גודל עץ השיעורים הנצפים הוא לכל היותר M-מספר השיעורים הכולל במבנה

אם נחבר את גודל כל המערכים שיש בטבלת הקורסים נקבל מערך גדול בגודל של כל השיעורים במבנה כלומר M.

כל סיבוכיות המקום של הפעולות הרקורסיוביות קטנות מסיבכות המקום שנדש על מנת לשמור את הנ"ל. לכן **סיבוכיות מקום כוללת** של המבנה:

O(M+N) כאשר M=מספר השיעורים הכולל במבנה וN מספר הקורסים הכולל המערכת.