

האובייקט Boom מורכב מהשדות הבאים :

## עץ קורסים ראשי

עץ הקורסים הראשי (בשם *main\_tree*) הוא עץ AVL שבו כל צומת מייצג אובייקט שהמפתח שלו הוא מזהה הקורס ואובייקט מטיפוס *lectureOfCourse*. לאובייקט *lectureOfCourse* יש את השדות הבאים :

- *courseID* - מזהה הקורס

- *numOfClasses* - מספר ההרצאות שיש בקורס הנוכחי

- *arr* - מערך הצבעה לרשימת הצפיות

עץ זה משמש לפעולות כמו למשל הוספת/הסרת קורס, הוספת זמן צפייה וכו' (יוסבר בהמשך).

## רשימת צפיות

רשימת הצפיות (בשם *priority\_list*) היא רשימה דו כיוונית המכילה מצביע לתחילת הרשימה ומצביע לסוף הרשימה (מטיפוס *NodeList*). כל איבר ברשימה מכיל את זמן הצפייה ובנוסף יהיה מצביע נוסף לאובייקט המכיל את כל ההרצאות בעלות זמן הצפייה הנוכחי (אסביר בהמשך על מבנה זה).

## עץ זמן צפיה אפס

עץ זמן צפיה אפס הוא עץ המכיל את כל הקורסים המכילים הרצאות עם זמן צפייה אפס. תחת כל קורס הגדרנו מזהה קורס ומערך של משתנים בוליאניים (*bool\_arr*) שמטרתו הוא ליידע עבור קורס מסוים לאילו הרצאות יש זמן צפייה אפס ולאילו לא. כלומר, במידה ו- $bool\_arr[i] = false$  אזי להרצאה ה-*i* יש זמן צפייה שונה מאפס ובמידה ו- $bool\_arr[i] = true$ , אזי להרצאה ה-*i* יש זמן צפייה אפס. בנוסף, לכל צומת בעץ קיים *counter* (בשם *false\_cells*) אשר אחראי לספור עבור קורס כלשהו לכמה הרצאות יש זמן צפייה שונה מאפס. ברגע ש-*false\_cells* שווה למספר ההרצאות של הקורס נמחק את הקורס מעץ זמן צפיה אפס. עץ זה משמש להכנסת מזהי הרצאות וקורסים בפונקציה *GetMostViewedClasses*.

## עץ קורסים תחת חוליית NodeList

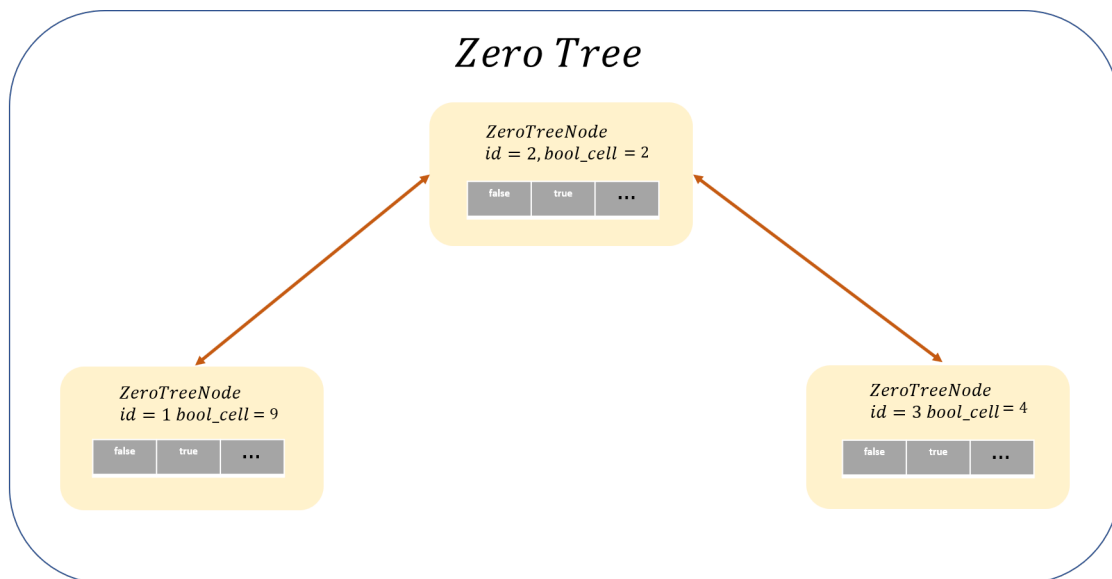
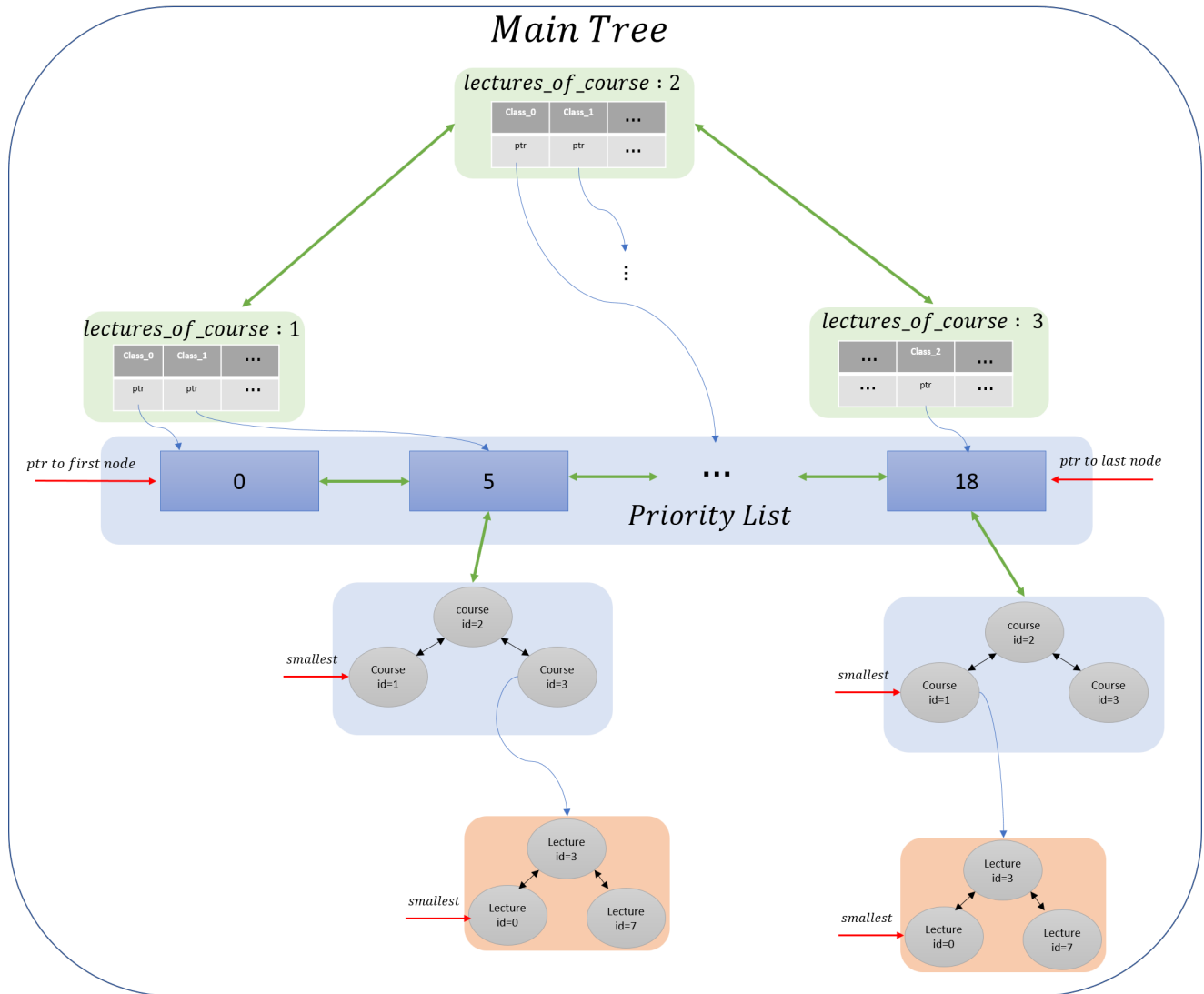
עץ הקורסים תחת חוליית *NodeList* הוא עץ כאשר המפתח (*key*) הוא מזהה הקורס והערך (*value*) הוא אובייקט מטיפוס *Course* המכיל מצביע לעץ המכיל את כל ההרצאות (*Lecture*) תחת הקורס הנוכחי בעלי זמן צפייה המשוך ל-*NodeList* הנוכחי. למעשה, בעץ זה אספנו את כל ההרצאות של קורס כלשהו עם זמן צפייה השייך ל-*NodeList* הנוכחי. אובייקט מטיפוס *Course* מכיל את השדות הבאים :

- מזהה הקורס

- עץ המכיל את כל ההרצאות

אובייקט מטיפוס *Lecture* מכיל את מזהה ההרצאה בלבד (המשויך לקורס הנוכחי).

בנוסף, לכל *lectureOfCourse* בעץ הקורסים הראשי השדה *arr* מכיל מצביעים ל-*NodeList* כאשר *arr[i]* מכיל מצביע ל-*NodeList* אשר תחתיה ההרצאה נמצאת (תחת הקורס המתאים).



## 2 מימוש הפונקציות

### 2.1 פונקציית *init*

- ראשית, ניצור מבנה  $DS$  - המכיל עץ קורסים ראשי ריק ורשימת צפיות המכילה  $NodeList$  המתאים ל-0 צפיות. בנוסף, ניצור עץ זמן צפייה אפס ריק.
  - במידה של כישלון, נחזיר  $NULL$ , אחרת נחזיר מצביע למבנה נתונים ריק.
- סיבוכיות זמן: יצירת 2 עצים ריקים ב- $O(1)$  ויצירת רשימה עם  $NodeList$  המתאימה ל-0 צפיות ב- $O(1)$ .

### 2.2 פונקציית *AddCourse*

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. במידה והקורס קיים נחזיר  $FAILURE$ .
- ניצור אובייקט מטיפוס  $ZeroTreeNode$  המכיל את מזהה הקורס ומערך של משתנים בוליאניים אשר מסמלים האם ההרצאה תחת האינדקס הנוכחי עם 0 צפיות או לא. לכן, באתחול נאתחל את איברי המערך להיות  $true$  - כלומר בהתחלה כל ההרצאות נמצאים עם זמן צפייה 0, כנדרש. לאחר מכן, נכניס איבר זה לעץ הקורסים  $zero\_tree$ .
- בנוסף, נקצה מערך הצבעה לחולייה ברשימת הצפיות המתאימה ל-0 צפיות ונעבור באיברי מערך  $arr$  ונגדיר כל תא להצביע ל- $NodeList$  שמתאים ל-0 צפיות.
- לאחר מכן, ניצור צומת בעץ הקורסים הראשי מטיפוס  $lectureOfCourse$  עם מזהה הקורס שניתן בפונקציה ועם מערך ההצבעה  $arr$ .

סיבוכיות זמן:

1. בדיקת שגיאות -  $O(1)$
  2. בדיקה האם הקורס הרצוי קיים -  $O(\log n)$
  3. יצירת אובייקט מסוג  $lectureOfCourse$  -  $O(1)$
  4. הכנסת האובייקט מסוג  $lectureOfCourse$  ותיקון איזון עץ הקורסים הראשי -  $O(\log n)$
  5. אתחול מערך ההצבעה מסוג  $NodeList^*$  -  $O(m)$
  6. יצירת אובייקט  $ZeroTreeNode$  ב- $O(m)$  (אתחול המשתנים הבוליאניים ל- $true$ ) והכנסתו לעץ  $zero\_tree$  ב- $O(\log n)$ .
- סה"כ סיבוכיות הזמן היא  $O(\log n + m)$ .

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. במידה ולא קיים הקורס נחזיר *FAILURE*.

- נמצא את הקורס המתאים לפי מזהה הקורס בעץ הקורסים הראשי.

– כעת עבור כל תא במערך *arr* נבצע את אוסף הפעולות הבא :

\* במידה וזמן הצפייה של החוליה אליה הוא מצביע שונה מאפס נבצע את הפעולות הבאות :

- נמצא את הקורס המתאים (ע"י גישה למצביע לחוליה המתאימה ברשימת הצפיות) שתחתיו נמצאת ההרצאה.
- נסיר את ההרצאה מהקורס המתאים.
- במידה ועץ ההרצאות ריק, נסיר את הקורס מעץ הקורסים אשר נמצא בחוליה המתאימה.
- במידה ולא נותרו עוד קורסים תחת החוליה המתאימה נסיר את החוליה המתאימה.

- לבסוף נמחק מ-*zero\_tree* את הקורס הנוכחי (במידה והקורס לא קיים ב-*zero\_tree*, לא יתבצע דבר).

- מסירים את הקורס (מטיפוס *lectureOfCoures*) מעץ הקורסים הראשי ונסיים את התוכנית.

#### סיבוכיות זמן :

- בדיקת פרמטרים -  $O(1)$ .
  - בדיקה האם הקורס הרצוי קיים בעץ הקורסים הראשי -  $O(\log n)$ .
  - עבור  $m$  הרצאות אנו :
    - מחפשים את הקורס של ההרצאה הנוכחית תחת החוליה המתאימה (לפי מערכת ההצבעה של *main\_tree*) -  $O(\log n)$
    - מסירים את ההרצאה תחת הקורס המתאים תחת החוליה הנוכחית -  $O(\log m)$
    - מסירים את הקורס תחת החוליה המתאימה במידה ובקורס לא נותרו עוד הרצאות -  $O(\log n)$
    - מסירים את החוליה במידה ולא נותרו קורסים תחת החוליה הנוכחית -  $O(1)$
  - הסרת הקורס (מטיפוס *ZeroTreeNode*) המתאים למזהה הקורס הרלוונטי מהעץ *zero\_tree*.
  - הסרת הקורס מעץ הקורסים הראשי -  $O(\log n)$ .
- סה"כ סיבוכיות הזמן היא  $O(m \log m)$ , ומכיוון ש- $m < M$  נסיק כי בפרט סיבוכיות הזמן היא  $O(m \log M)$ .

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. במידה ולא קיים הקורס נחזיר *FAILURE*.
- נמצא את הקורס המתאים בעץ הקורסים הראשי וניגש לחוליה המתאימה דרך מערך המצבעים *arr* ברשימת הצפיות תחתיה קיים הקורס המתאים עם ההרצאה המתאימה.
- במידה וזמן הצפייה של החוליה שהגענו אליה הוא 0 - ניגש ל-*zero\_tree*, נחפש את הקורס הרלוונטי וניגש במערך המשתנים הבוליאניים לאינדקס *classID* ונשנה את ערכו ל-*false*.
- \* במידה ו-*false\_cells* שווה למספר ההרצאות של קורס זה, נסיר את הקורס הנוכחי מ-*zero\_tree* כי במקרה זה כל זמני הצפייה של ההרצאות תחת הקורס הנוכחי שונים מאפס.
- אחרת, נסיר את ההרצאה מהעץ של הקורסים תחת החוליה הנוכחית. במידה וזוהי ההרצאה היחידה תחת העץ של הקורס הרלוונטי נמחק את הקורס המשוויך לחוליה המתאימה. במידה ועץ הקורסים תחת החוליה הוא ריק נמחק את החוליה ונשרש בהתאם את רשימת הצפיות.
- כעת נעבור על רשימת הצפיות החל מהחוליה בה אנו נמצאים עד אשר נגיע לחוליה שזמן הצפייה (*key*) שלה גדול או שווה לזמן הצפיות החדש של ההרצאה (זהו הסכום של הזמן הנוכחי עם הזמן שניתן כפרמטר לפונקציית *WatchClass*). נשים לב ש- $time > 0$  ולכן לא יכולה להתבצע הכנסה של הרצאה תחת חוליה המתאימה לזמן צפייה 0.
- במידה ויש חוליה המשוויכת לזמן החדש המתאים נחפש את הקורס המשוויך להרצאה. במידה והקורס קיים, נוסיף את ההרצאה תחת עץ הקורס המשוויך לחוליה המתאימה. במידה והקורס לא קיים, נבצע הוספת קורס עם מזהה הקורס הרלוונטי ונכניס לעץ שלו את ההרצאה הדרושה.
- במידה ולא קיימת חוליה המשוויכת לזמן החדש נקצה חוליה חדשה עם הזמן המתאים, נשרש אותה לרשימת הצפיות ונכניס קורס חדש שתחת עץ ההרצאות נכניס את ההרצאה הדרושה.

#### סיבוכיות זמן:

1. בדיקת פרמטרים -  $O(1)$ .
  2. בדיקה האם הקורס הרצוי קיים -  $O(\log n)$ .
  3. גישה לחוליה ברשימת הצפיות ב- $O(1)$ , גישה לקורס תחת החוליה המתאים ב- $O(\log n)$  ולאחר מכן גישה להרצאה תחת הקורס המתאים ב- $O(\log m)$ .
  4. מעבר על כלל היותר  $t$  חוליות ברשימת הצפיות ב- $O(t)$ .
  5. הוספת חוליה ברשימת הצפיות במידה וזמן הצפייה המתאים לא קיים ב- $O(1)$ .
  6. חיפוש הקורס בעץ הקורסים תחת החוליה המתאימה ב- $O(\log n)$ .
  7. במידה ולא קיים הקורס מבצעים הכנסה של קורס ב- $O(\log n)$ .
  8. הכנסת הרצאה תחת הקורס המתאים תחת החוליה המתאימה ב- $O(\log m)$ .
- סה"כ כי סיבוכיות הזמן היא  $O(\log m + t)$ , ומכיוון ש- $m < M$  נסיק כי בפרט סיבוכיות הזמן היא  $O(\log M + t)$ , כנדרש.

## 2.5 פונקציית *TimeViewed*

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. נחפש את הקורס (מטיפוס *lectureOfCourse*) - במידה ולא קיים הקורס נחזיר *FAILURE*.

- אחרת, ניגש במערך המצביעים לאינדקס ה-*classID* וניגש לערך המפתח של ה-*NodeList* אליו הוא מצביע.

- כעת, נשנה את ערכו של *\*timeViewed* לערך המפתח של ה-*NodeList*, כנדרש.

- נחזיר ערך הצלחה.

### סיבוכיות זמן:

1. בדיקת פרמטרים -  $O(1)$ .
  2. בדיקה האם הקורס הרצוי קיים -  $O(\log n)$ .
  3. גישה במערך של הקורס במידה ומעבר לחולייה ברשימת הצפייה דרך המצביע המתאים (לפי *classID*) -  $O(1)$ .
  4. שינוי הפרמטר שניתן בפונקציה לערך המפתח לחולייה אליה הגענו -  $O(1)$ .
- סה"כ סיבוכיות הזמן היא  $O(\log n)$ , כנדרש.

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. נחפש את הקורס (מטיפוס *lectureOfCourse*) - במידה ולא קיים הקורס נחזיר *FAILURE*.

- נעבור על רשימת הצפיות מזמן הצפייה הגבוהה לנמוך ע"י גישה לאיבר שמצביע לחוליה בעלת זמן הצפייה הגדול ביותר ברשימת הצפיות. בכל חוליה נכנס לעץ הקורסים וניגש לקורס בעל מזהה הקורס הנמוך ביותר כאשר עץ *AVL* תומך בהחזרת איבר בעל מזהה הקורס הנמוך ביותר.

— במידה וגובה עץ הקורסים הנוכחי גדול מ- $m$ , נעלה מהצומת הקטן ביותר  $m$  צמתים במעלה העץ ונבצע סיור *In Order* עד שעברנו על  $m$  קורסים. עבור כל קורס נכנס לעץ ההרצאות שלו, ניגש להרצאה עם המזהה הקטן ביותר (ניגש ל-*smallest* - לפי המימוש).

\* במידה וגובה עץ ההרצאות גדול מ- $m$ , נעלה מהצומת הקטן ביותר  $m$  צמתים העץ במעלה ונבצע סיור *In Order* עד שעברנו על  $m$  הרצאות. כעת נשמור את ההרצאות במערך *classes* לפי הסדר כאשר ערכי ה-*courses* במערך המתאים לאינדקס יהיה מזהה הקורס תחתיו אנו נמצאים.

\* במידה וגובה העץ קטן מ- $m$ , נעלה לשורש העץ הכנסה של האיברים המתאימים למערכים *course*-ו *classes* באמצעות סיור *In Order*, אם הגענו להכנסה של  $m$  איברים למערכים, נסיים את התוכנית. אחרת, נעבור לקורס הבא.

— במידה וגובה העץ קטן מ- $m$  נעלה לשורש מהצומת הקטן ביותר במעלה העץ ונבצע סיור *In Order*. עבור כל קורס נכנס לעץ ההרצאות שלו, ניגש להרצאה עם המזהה הקטן ביותר (ניגש ל-*smallest* - לפי המימוש).

\* במידה וגובה עץ ההרצאות גדול מ- $m$ , נעלה מהצומת הקטן ביותר  $m$  צמתים במעלה ונבצע סיור *In Order* עד שעברנו על  $m$  הרצאות. כעת נשמור את ההרצאות במערך *classes* לפי הסדר כאשר ערכי ה-*courses* במערך המתאים לאינדקס יהיה מזהה הקורס תחתיו אנו נמצאים.

\* במידה וגובה עץ זה קטן מ- $m$ , נעלה לשורש העץ ונבצע הכנסה של האיברים למערכים, אם הגענו להכנסה של  $m$  איברים למערכים נסיים את התוכנית. אחרת, נעבור לקורס הבא.

— במידה והגענו ל- $m$  הכנסות למערכים, נסיים את הריצה. אחרת, נעבור לחפש בעץ הקורסים של החוליה הקודמת.

- במידה והגענו לחוליה המתאימה לזמן צפייה 0, נעבור לעץ *zero\_tree* וניגש לאיבר *smallest*. באופן דומה נעלה לכל היותר  $m$  צמתים או מקסימום עד השורש, ונבצע סיור *In Order* על איברי העץ (*ZeroTreeNode*).

— כעת נבצע את הפעולות הבאות עד אשר הגענו להכנסת  $m$  איברים למערכים *classes*-ו *courses*: נעבור על איברי המערך *bool\_arr*. במידה והגענו לערך *true* נבצע הכנסה למערכים *classes*-ו *courses* עם מספר ההרצאה (זהו למעשה האינדקס הנוכחי של המערך) ועם מספר הקורס (מזהה הקורס הנוכחי).

#### סיבוכיות זמן:

1. בדיקת פרמטרים -  $O(1)$ .
2. מעבר על רשימת הצפיות מהסוף להתחלה תוך גישה בכל חוליה למזהה הקורס הנמוך ביותר (ב- $O(1)$ ), עלייה במעלה עץ הקורסים תחת החוליה הנוכחית וסיור *inOrder* כאשר בכל קורס מבצעים דבר דומה עבור ההרצאות עד מעבר על  $m$  הרצאות -  $O(m)$ .
3. במידה ומגיעים לחוליה של זמן צפייה אפס נעבור ל-*zero\_tree*. ניגש לאיבר *smallest* ב- $O(1)$ . נעלה  $m$  צמתים או מקסימום עד השורש ונבצע סיור *InOrder* על איברים מטיפוס *ZeroNodeTree* כאשר בכל אחד מהם יש מערך של משתנים בוליאניים. עבור כל מערך סורקים כל תא במערך. במידה ונתקל בתא שערכו הוא *true* אז נכניס את הערכים המתאימים למערכים. נשים לב שכל תא במערך שערכו *false* הוא איבר שזמן הצפייה שלו איננו אפס, כלומר כבר עברנו עליו מקודם ולכן לכל היותר אנו עוברים על כל הרצאה מקסימום פעמיים ולכן סיבוכיות הזמן היא  $O(m)$ .

סה"כ סיבוכיות הזמן היא  $O(\log n)$ , כנדרש.

## 2.7 פונקציית Quit

- ראשית, נמחק את עץ הקורסים הראשי אשר יסיר כל צומת ובין היתר נשחרר את מערך המצביעים.
- נמחק את עץ זמן צפייה אפס ואת כל האיברים שבו.
- נבצע עבור כל חוליה ברשימת הצפיות את סט הפקודות הבא:
  - עבור כל קורס בעץ הקורסים תחת החוליה הנוכחית נמחק את עץ ההרצאות שלו ונמחק את הקורס הנוכחי.
  - נשחרר את עץ הקורסים הקשור לחוליה הרלוונטית.
- לבסוף נמחק את רשימת הצפיות.
- נבצע  $DS = nullptr$  ונסיים את התוכנית.

### סיבוכיות זמן:

- מחיקת עץ הקורסים הראשי - מעבר על  $n$  אובייקטים מסוג `lectureOfCourse` -  $O(n)$ .
- שחרור כל האובייקטים מטיפוס `Course` תחת רשימת הצפייה כאשר בכל קורס נשחרר את עץ ההרצאות תחתיו - יש לכל היותר  $m$  הרצאות ו- $n$  קורסים ולכן סה"כ -  $O(n + m)$ .
- שחרור רשימת הצפיות - יש לכל היותר  $m$  חוליות (כמספר ההרצאות לכל היותר) ולכן סה"כ -  $O(m)$ .
- מחיקת עץ זמן הצפייה ואת האיברים שבו - לכל היותר  $O(m)$ .
- סה"כ סיבוכיות הזמן היא  $O(n + m)$ .

## 3 סיבוכיות מקום

במבנה שבנינו קיימים:

- עץ הרצאות ראשי - לכל היותר  $n$  איברים מטיפוס `lectureOfCourse`.
- לכל איבר מטיפוס `lectureOfCourse` יש מערך מצביעים כאשר מספר התאים עבור כל הקורסים הוא  $m$ .
- רשימת צפיות עם לכל היותר  $m$  חוליות.
- עצי קורסים תחת החוליה הרלוונטית - יש לכל היותר  $m$ .
- עצי הרצאות תחת כל קורס - יש לכל היותר  $m$  עצים כאלו.
- עץ זמן צפייה אפס כאשר יש לכל היותר  $n$  קורסים ולכל היותר יש  $m$  משתנים בוליאניים בכל הקורסים תחת עץ זה.

סה"כ סיבוכיות המקום היא  $n + m + m + m + m + n + m = O(n + m)$