

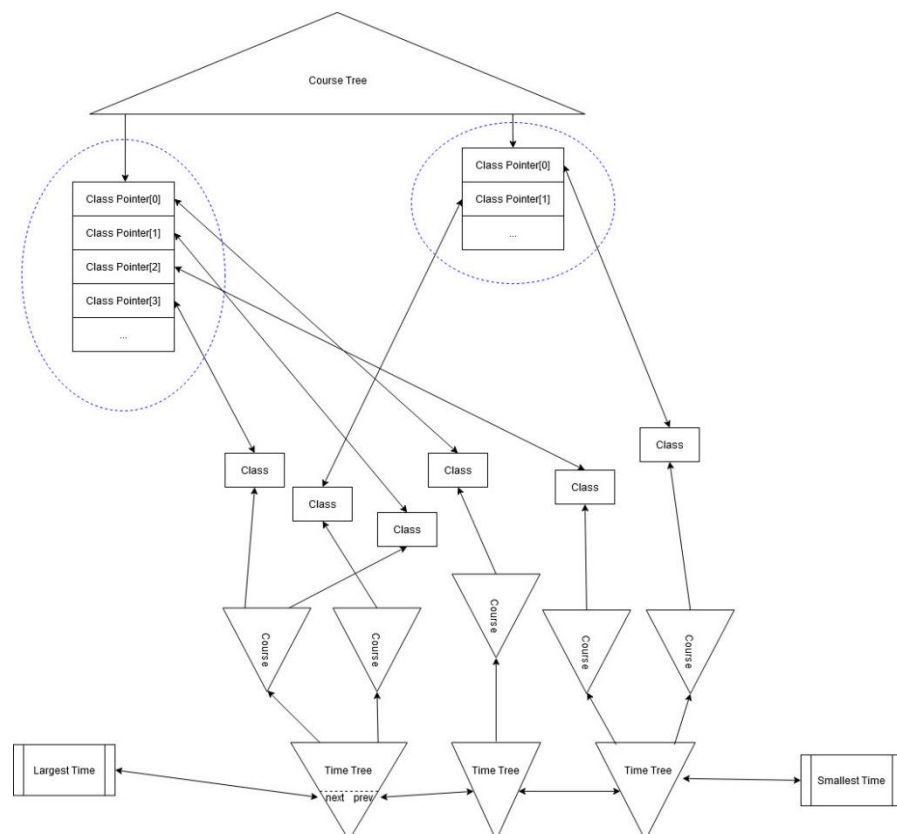
רטוב 1 – חלק יבש

תיאור כללי:

מבנה הנתונים מכיל עץ קורסים, כל איבר של קורס מכיל מערך של מצביעים של שיעורים. בנוסף המבנה מכיל רשימה מקושרת, כך שכל איבר i ברשימה תפקידו להכיל את כל השיעורים שלהם i צפיות. למעשה איבר i ברשימה המקושרת מכיל איברים, כך שכל איבר פנימי j הוא עץ שיעורים, כך שכל השיעורים תחת אותו העץ j שייכים לאותו הקורס j . בנוסף נציין כי איברי השיעורים מוצבעים על ידי האיברים במערכים שפורטו מקודם.

מבנה הנתונים מכיל:

- Course Tree - זהו עץ AVL, שמכיל בכל איבר מערך של מצביעים לשיעורים של קורס מסויים. המפתח שלו הוא מספר הקורס כפי שהוכנס בהוספת קורס.
- TimeList - זוהי רשימה מקושרת דו-כיוונית. תפקיד כל איבר בעל אינדקס i ב $TimeList$, הוא להכיל את כל השיעורים שזמן הצפייה בהם i דקות, ופנימית הם מסודרים בעצים מסוג $SubTreeCourse$. איברי $SubTreeCourse$ מסודרים כעץ AVL כאשר המפתח שלהם הוא מספר הקורס.
- SubTreeCourse - עצים מסוג זה הם עצי AVL המוכללים באיבר ברשימה דו-כיוונית ($TimeList$), כל עץ j מסוג זה, מכיל שיעורים שלהם i דקות צפייה ושייכים לקורס j . כל השיעורים מסודרים בעץ לפי מספר השיעור. עצים אלו מכילים איבר מסוג $Class$.
- Class - איברים אלו מסודרים בעצים מסוג $SubTreeCourse$ וגם מוצבעים על ידי מערכי הקורסים בעץ $CourseTree$. הם מייצגים את השיעורים עצמם.
- LargestTime - איבר זה הוא מצביע לאיבר הגדול ב- $TimeList$.
- SmallestTime - איבר זה הוא מצביע לאיבר הקטן ב- $TimeList$.



ניתוח סיבוכיות מקום:

סיבוכיות המקום של המבנה היא $O(m+n)$, כאשר m הוא סך כל השיעורים במבנה ו- n מספר הקורסים.

מספר האיברים ב-CourseTree הוא n מכיוון שישנם n קורסים.

מספר האיברים בתוך כלל מערכי הקורסים הוא מספר השיעורים m מכיוון שיש מצביע בודד לכל שיעור.

מספרי איברי class הוא m , מכיוון שלכל שיעור מתאים איבר class יחיד.

מספר עצי CourseTree, הוא לכל היותר m , מכיוון שכל עץ כזה מכיל לפחות איבר class אחד.

מספר האיברים (עצים) ב-TimeList הוא לכל היותר m , מכיוון שלא יכול להיות שיש איבר שלא מכיל אף שיעור.

בנוסף ישנם שני איברים בודדים LargestTime, SmallestTime.

$$\text{סה"כ: } n + 4m + 2 = O(n + m)$$

מימושים לפונקציות וניתוח סיבוכיות זמן:

1. void* Init() – אתחול שלושה איברים בודדים, LargestTime, SmallestTime ושורש ריק בעץ CourseTree. סה"כ אתחול איברים בודדים ב- $O(1)$

2. addCourse(*DS, courseID, numOfClasses) – מייצרים m איברי Class – $O(m)$. מייצרים מערך מצביעים בגודל m , כך שכל איבר במערך מצביע לאחד מאיברי Class לפי האינדקס שלהם – $O(m)$. את המערך הזה נכניס כאיבר לעץ CourseTree – $O(\log(n))$.

בנוסף נייצר עץ מסוג SubTreeCourse באופן הבא:

נבנה עץ כמעט שלם עם m איברים ממוינים לפי המערך כפי שראינו בתרגיל בתרגול – $O(m)$. את העץ שבנינו נכניס לעץ המוצבע מאיבר TimeList בעל האינדקס 0 (אם SmallestTime גדול מ-0, ניצור איבר חדש ב-TimeList בעל אינדקס 0 ו-SmallestTime יצביע עליו) – $O(\log(n))$ לפי מספר הקורסים המקסימלי שיכול להיות באיבר רשימה - TimeList.

$$\text{סה"כ } 2 * O(m) + 2 * O(\log(n)) = O(m + \log(n))$$

3. RemoveCourse(*DS, CourseID) – ניגש למערך המצביעים של הקורס דרך עץ CourseTree – $O(\log(n))$. נעבור איבר איבר במערך ונבצע את הפעולות הבאות:

- ניגשים בעזרת המצביע לאיבר Class – $O(1)$.
- נמחק את איבר זה מעץ SubTreeCourse – $O(\log(m))$. אם זהו האיבר האחרון ב-SubTreeCourse נמחק את SubTreeCourse מאיבר הרשימה ב-TimeList – $O(\log(M))$.

נסיר את המערך שהוא איבר ב-CourseTree – $O(\log(n))$.

$$\text{סה"כ - } 2 * O(\log(n)) + O(m * \log(M)) = O(m * \log(M))$$

4. $WatchClass(*DS, CourseID, ClassID, time)$ –

ניגש למערך המצביעים של הקורס הרצוי – $O(\log(n))$.

ניגש דרכו לאיבר $ClassID$ הרצוי – $O(1)$

נוציא את איבר $ClassID$ מעץ $SubTreeCourse$ – $O(\log(M))$. אם זהו האיבר האחרון ב-
 $SubTreeCourse$ נמחק את $SubTreeCourse$ מאיבר $TimeList$ – $O(\log(n))$. אם לאחר מכן העץ של
איבר $TimeList$ ריק, נמחק את איבר הרשימה $O(1)$.

נעבור על הרשימה עד להגעה לאינדקס שגדול או שווה ל- $i+time$ או לסוף הרשימה – $O(t)$.

במידה ולא קיים צומת רשימה בעל אינדקס $i+time$ נוצר כזה ונכניס לרשימה – $O(1)$. במידה ולא
קיים $SubTreeCourse$ לקורס הרצוי באיבר הרשימה הנוכחי, נוצר כזה – $O(1)$ ונוסיף אותו לעץ
שמוחזק בצומת הרשימה – $O(\log(n))$. בעץ זה נכניס את איבר $ClassID$ – $O(\log(M))$. במידת הצורך
נעדכן $LargestTime \setminus SmallestTime$ – $O(1)$.

סה"כ $5 * O(1) + 3 * O(\log(n)) + 2 * O(\log(M)) + O(t) = O(t + \log(M))$

5. $TimeViewed(*DS, CourseID, ClassID, *timeViewed)$ – ניגש לאיבר $ClassID$ הרצוי דרך
הקורס שנמצא ב- $CourseTree$ – $O(\log(n))$.

ניגש לאיבר הרשימה ב- $TimeList$ דרך מצביע הנמצא ב- $ClassID$ – $O(1)$. נאחסן את הערך של
איבר הרשימה במצביע החזרה.
סה"כ $O(\log(n))$.

6. $GetMostViewedClass(*DS, numOfClasses, *Courses, *Classes)$ – ניגש דרך
 $LargestTime$ לאיבר בעל האינדקס הגדול ביותר ברשימה – $O(1)$.

נעבור בסיוור $inorder$ הפוך (מהגבוה לנמוך) על עצי $SubTreeCourse$. ובתוך כל עץ נבצע
סיוור $inorder$ נוסף. אם הגענו למספר הרצוי נפסיק. במקרה שסיימנו לעבור על כלל
האיברים באיבר הרשימה הנוכחי, נבצע את אותה פעולה על האיבר שאחריו. לכל איבר
 $class$ אנחנו מבצעים פעולות ב- $O(1)$. בסה"כ אנו עוברים על m איברים ולכן הסיבוכיות היא
 $O(m)$.

הערות:

- בעת סיוור $inorder$ בעץ AVL , סיבוכיות הזמן היא $\min\{O(m), O(n)\}$ כאשר m מספר האיברים
בעץ, n הוא מספר האיברים הנדרש. גם אם נרצה לעבור על n איברים הנמצאים בכמה
עצים אשר ממוינים בינם לבין עצמם, הפעולה תתבצע ב- $O(n)$ מכיוון שמספר העצים אינו
משנה אלא רק מספר האיברים, מפני שלא יכולים להיות יותר עצים מאשר איברים.

7. $Quit(**DS)$ –

בדומה להדפסה נעבור על איברי $ClassID$ מהגדול לקטן ונמחק איבר איבר. כאשר מוחקים איבר אחרון
בעץ $SubTreeCourse$ נמחק גם אותו מהעץ העליון. אם זהו $subTreeCourse$ האחרון בעץ נמחק גם
את העץ ואץ צומת הרשימה $TimeList$, אין צורך בגלגולים והמעבר הוא $inorder$.

סה"כ עוברים על M איברי $Class$, M עצי $SubTreeCourse$ לכל היותר, M איברי $TimeList$ לכל היותר
סה"כ $O(3M)$.

כעת נעבור בסיור $inorder$ על עץ $CourseTree$ ונמחק איבר איבר, סה"כ $O(n)$.

לכן הסיבוכיות הכוללת של הפונקציה היא $O(n+M)$.