

הצגת מבנה הנתונים:

מבנה הנתונים שלנו מורכב משלושה עצי AVL ושתי רשימות מקושרות דו כיוונית.

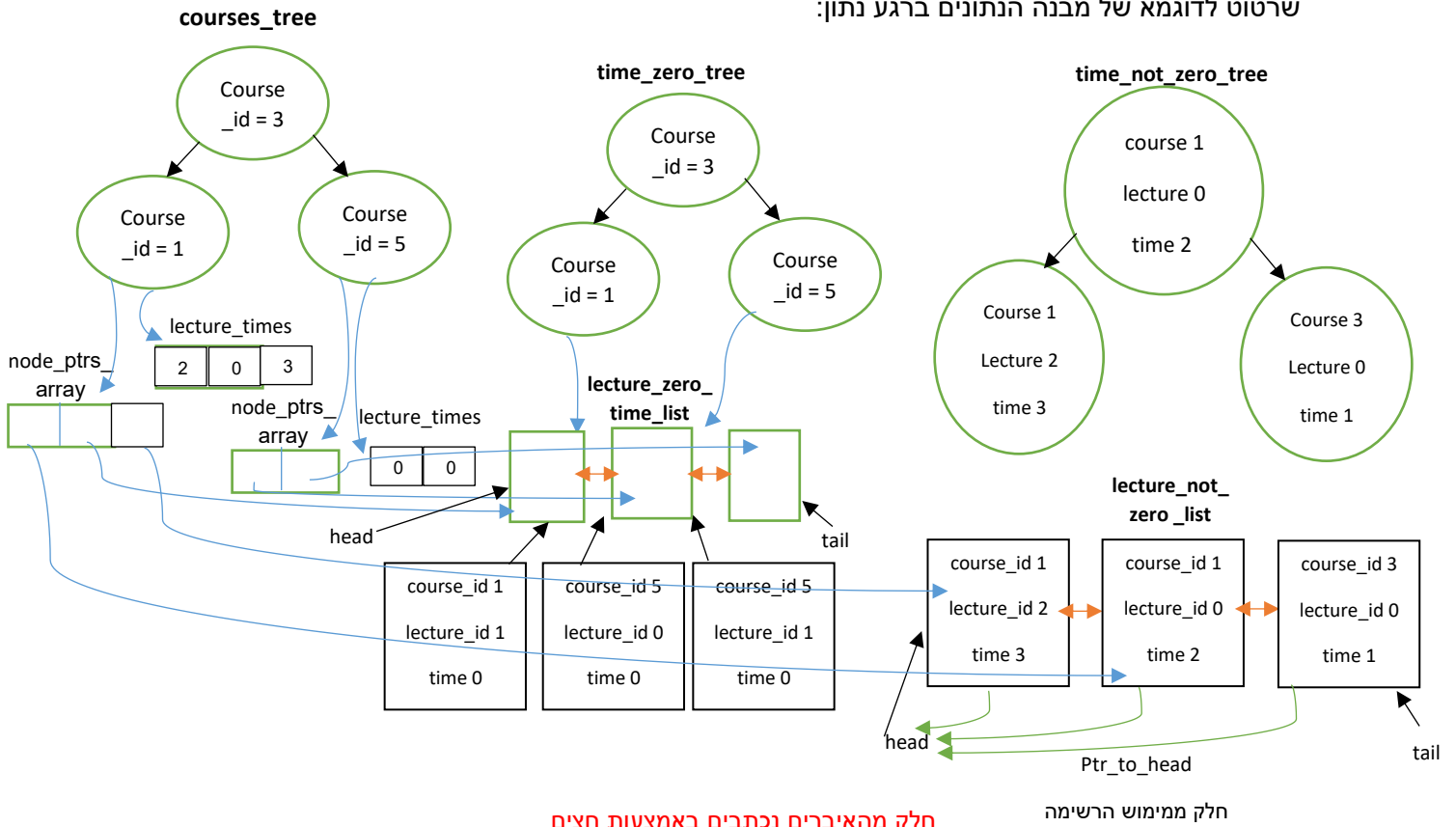
נתחיל מפירוט האובייקטים בהם השתמשנו לייצוג המידע במבנה הנתונים:

- Course – אובייקט המייצג קורס. כל אובייקט שכזה מחזיק במספר ID ייחודי, מספר ההרצאות בעלות זמן צפייה כולל אפס עד כה, ושני מערכים. האחד מערך המחזיק את זמני הצפייה בכל הרצאה של הקורס, lecture_times, והשני, node_ptrs_array, מערך המחזיק מצביעים לאובייקטים מטיפוס Node<Lecture> עליהם נסביר בקרוב.
- CourseWithoutTime – אובייקט של קורס המחזיק רק ID ומצביע ל-Node<Lecture>. אובייקט זה בא בשימוש בעץ המחזיק קורסים להם יש הרצאות עם זמן צפייה אפס. נפרט בהרחבה בהמשך.
- Lecture – אובייקט המייצג הרצאה. כל אובייקט שכזה מחזיק מספר ID של הקורס לו שייכת ההרצאה, מספר ID של ההרצאה (אשר ייחודי לה מבין הרצאות אותו הקורס), ואת זמן הצפייה בהרצאה זו.

כעת נוכל לפרט בהרחבה אודות מבני הנתונים העיקריים (שרטוט מצורף מטה):

1. עץ AVL של קורסים – courses_tree: בכל צומת בעץ אנו מחזיקים אובייקט מטיפוס Course. כפי שהוסבר, לכל צומת בעץ מספר ID של קורס במערכת הקורסים שלנו, מערך המייצג זמני צפייה בהרצאות ומערך מצביעים. כל מצביע באינדקס i במערך שכזה מתאים להרצאה ה-i של הקורס. המצביע ה-i מצביע על ההרצאה ה-i של הקורס, באחת הרשימות: הרשימה המייצגת הרצאות עם זמן אפס, או ברשימה המייצגת זמן גדול מאפס. עוד בהמשך.
2. עץ AVL של קורסים להם הרצאות עם זמן צפייה אפס – time_zero_tree: כל צומת בעץ זה היא אובייקט מטיפוס CourseWithoutTime, ממנו יש מצביע להרצאה האחרונה השייכת לקורס זה ברשימה המייצגת את העץ הנ"ל. נפרט כעת לגבי רשימה זו.
3. רשימה מקושרת דו-כיוונית של הרצאות – lecture_zero_time_list: רשימה מקושרת בה כל איבר הוא הרצאה, וההרצאות מסודרות בסדר כרונולוגי עולה כאשר ההשוואה נעשית ע"פ קורס ID והרצאה ID כפי הנדרש בתרגיל. נובע מכך ישירות שלכל קורס ולו מספר הרצאות בעלות זמן צפייה אפס, הרצאות אלו יופיעו בזו אחר זו ברשימה הנ"ל. לכל הרצאה אחרונה ברשימה של קורס k, כלומר ההרצאה עם ה-ID הגבוה ביותר מבין ההרצאות בעלות זמן צפייה אפס שלו, נחזיק מצביע להרצאה זו מהקורס ה-k בעץ time_zero_tree. לגבי עדכון המצביעים תוך הסרה והכנסה, הכנסת ההרצאות במיקום הנכון בעת הוספת קורס, נפרט בהמשך.
4. עץ AVL של הרצאות בעלות זמן צפייה גדול מאפס – time_not_zero_tree: ההשוואה בעץ נעשית כפי הנדרש בתרגיל, ראשית ע"פ זמן וכן הלאה. בכל צומת אובייקט Lecture.
5. רשימה מקושרת דו כיוונית של הרצאות – lectures_not_zero_list: זו רשימה אשר למעשה מתרגמת את העץ time_not_zero_tree. לגבי הכנסה לרשימה, הזזה של איברים בה בעת ביצוע WatchClass נפרט בהמשך. כעת נציין רק כי על כל הרצאה i של קורס k ברשימה, קיים מצביע במערך מצביעים של הקורס k, כך שהמצביע באינדקס i במערך מצביע על ההרצאה הנ"ל ברשימה (כאמור במידה והיא בעלת זמן צפייה גדול מאפס, שכן אחרת הייתה ברשימה בעלת זמן צפייה אפס).

שרטוט לדוגמא של מבנה הנתונים ברגע נתון:



חלק מהאיברים נכתבים באמצעות חצים
ע"מ שלא להיאלץ לכתוב קטן מידי,
מפאת חוסר מקום, לדוגמא הרצאות

חלק ממימוש הרשימה

פעולות על מבני הנתונים:

עץ: בעץ קיימות הפעולות הסטנדרטיות (הכנסה, הוצאה) שנכתבו לפי האלגוריתם שניתן בהרצאות, ולכן סיבוכיות פעולות אלו הן $O(\log n)$. בנוסף, קיימות הפונקציות הבאות:

find: מוצאת את האיבר הניתן לה ומחזירה את הצומת עם מידע זה. מחזירה NULL אם אין צומת כזה.

סיבוכיות: הפעולה מבצעת חיפוש בעץ בינארי (AVL) ולכן לוקחת $O(\log n)$, מכיוון שגובה העץ הוא $\log n$.

findPreviousInOrder: מוצאת את האיבר הקודם לאיבר הניתן לה, ומחזירה NULL אם זה האיבר הקטן ביותר.

סיבוכיות: הפעולה מבצעת חיפוש בעץ בינארי (AVL) לחיפוש האיבר, ומוצאת את האיבר הקודם לו (ע"י מציאת האיבר המקסימלי בעץ השמאלי של הצומת עם המידע, או אם לא קיים עץ כזה ע"י שמירת הצומת האחרונה בה ירדנו ימינה) ולכן לוקחת $O(\log n)$, מכיוון שגובה העץ הוא $\log n$.

רשימה מקושרת דו-כיוונית: ברשימה קיימות פעולות ההכנסה הסטנדרטיות (בתחילת הרשימה, בסוף הרשימה ואחרי איבר נתון) שנכתבו לפי האלגוריתם שניתן בהרצאות (שינוי המצביעים של האיבר הקודם והאיבר הבא אחרי האיבר שהוכנס), וכן פעולת הסרה (שגם בה מתבצע שינוי מצביעים בהתאם). כפי שראינו בהרצאה, מדובר במספר סופי של פעולות לכל הכנסה והוצאה ולכן סיבוכיות פעולות אלו הן $O(1)$.

- מעתה בהסברים כאשר נתייחס לעץ המכיל קורסים להם הרצאות עם זמן צפייה אפס, נכתוב העץ של $time=0$, וכאשר נתייחס לעץ ההרצאות עם זמן צפייה גדול מאפס, נכתוב העץ של $time \neq 0$.
- נעשה את אותו דבר בהתייחסות לרשימות.

הפעולות הנדרשות:

:Init

ניצור אובייקט חדש מסוג CoursesManager. ישנם 5 מבני נתונים באובייקט זה:

- עץ קורסים ריק.
- עץ $time=0$ ריק.
- רשימה של העץ $time=0$ ריקה.
- עץ $time \neq 0$ ריק.
- רשימה של העץ $time \neq 0$ ריקה.

נדרש לאתחל כל מבנה נתונים. כל אתחול של מבנה נתונים, עץ ורשימה, לוקח $O(1)$, (שכן ללא איברים כעת) ולכן סך הכל הסיבוכיות היא $O(1)$.

AddCourse

ראשית נבדוק תקינות קלט כדרוש בתרגיל. לאחר מכן נבדוק האם הקורס קיים בעץ הקורסים $O(\log(n))$.

1. נוסף את הקורס החדש לעץ הקורסים $(O(\log n))$, ונפתח לו שני מערכים בגודל m : מערך של זמנים המאותחל ל-0 לכל הרצאה של קורס זה $(O(m))$, וכן מערך של מצביעים לאיברי רשימה מקושרת דו-כיוונית $(O(m))$.
2. נוסף את הקורס החדש לעץ $time=0$ $(O(\log n))$, נמצא את הקורס הקודם לו בעץ $time=0$ $(O(\log n))$, ובזכות המצביע להרצאה האחרונה ברשימה של העץ $time=0$ עבור קורס זה, נמצא מאיפה להתחיל להכניס את כל ההרצאות של הקורס החדש.
3. נכניס את כל m ההרצאות לרשימה של העץ $time=0$, בזו אחר זו לפי סדר עולה, תוך עדכון המערך של המצביעים של כל הרצאה $(O(m))$.
4. כשנגיע להרצאה האחרונה, נעדכן מצביע אליה מהקורס בעץ $time=0$.

הסיבוכיות הכוללת היא $2 \log(n) + 2m + \log(n) + \log(n) + m + 1 = 4 \cdot \log(n) + 3 \cdot m + 1$
ולכן סך הכל הסיבוכיות היא $O(\log(n) + m)$.

RemoveCourse

ראשית נבדוק תקינות קלט כדרוש בתרגיל. לאחר מכן נבדוק האם הקורס קיים בעץ הקורסים $O(\log(n))$.

1. נמצא את הקורס בעץ הקורסים $(O(\log n))$.
2. נעבור על מערך ההרצאות של הקורס $(O(m))$, ונסיר כל איבר של הרצאה מהרשימה המתאימה לו (רשימה של העץ $time=0$ או רשימה של העץ $time \neq 0$) $(O(1))$.
- 2.1 במידה והצבענו על איבר ברשימה של $time \neq 0$, נסיר אותו מן העץ $time \neq 0$ $(O(\log M))$.
3. נסיר את הקורס מן העץ של $time=0$ במידה והייתה לו לפחות הרצאה אחת שלה זמן צפייה אפס $(O(\log n))$.
4. נסיר אותו מעץ הקורסים $(O(\log n))$.

הסיבוכיות הכוללת היא: $2 \log(n) + 1 + m \log(M) + \log(n) + \log(n) = 4 \cdot \log(n) + m \log(M) + 1$
מכיוון ש- $M \gg n$, סך הכל הסיבוכיות היא $O(m \log(M))$.

:WatchClass

ראשית נבדוק תקינות קלט כדרוש בתרגיל. לאחר מכן נבדוק האם הקורס קיים בעץ הקורסים $O(\log(n))$. במידה והקורס קיים נבדוק חוקיות מספר ההרצאה. במקרה הצורך נחזיר שגיאה מתאימה.

1. נמצא את הקורס בעץ הקורסים $(O(\log n))$.
2. ניגש למקום ההרצאה ברשימה המתאימה לה בזכות מערך המצביעים $(O(1))$. נחלק כעת למקרים:
 - 2.1. להרצאה זו $time=0$, וזו ההרצאה האחרונה של הקורס לה $time=0$: נוציא את ההרצאה מהרשימה באמצעות המצביע מסעיף 2 $(O(1))$, ולאחר מכן נוציא את הקורס מן העץ $time=0$ $(O(\log n))$.
 - 2.2. להרצאה זו $time=0$, ומלבד הרצאה זו יש לקורס עוד לפחות הרצאה אחת עבודה $time=0$: נמצא את הקורס בעץ $time=0$ $(O(\log n))$ ונבדוק האם זו ההרצאה במיקום האחרון, כלומר ההרצאה האחרונה ברשימה של העץ $time=0$ עבור קורס זה. נזכיר כי אני מחזיקים לכל צומת בעץ $time=0$ מצביע להרצאה האחרונה שלו ברשימה ולכן $(O(1))$. במידה וכן, נעדכן את המצביע של ההרצאה האחרונה עבור $time=0$ לאיבר לפניו ברשימה $(O(1))$ ונסיר את ההרצאה מהרשימה $(O(1))$.
 - 2.3. להרצאה זו $time \neq 0$: נסיר מהעץ $time \neq 0$ את ההרצאה $(O(\log M))$ וכן מהרשימה של העץ $time \neq 0$ $(O(1))$.
 3. נעדכן את הזמן במערך הזמנים של הקורס, $lecture_times$, עבור ההרצאה $(O(1))$.
 4. נוסיף את ההרצאה עם הזמן המעודכן לעץ $time \neq 0$ $(O(\log M))$.
 5. נמצא את ההרצאה שלפני ההרצאה שהכנסנו בעץ $time \neq 0$. לפשטות ההסבר נקרא לה $prev_lecture$ (זוהי ההרצאה שאחריה ההרצאה החדשה, לאחר עדכון הזמן, צריכה להיכנס ברשימה של $time \neq 0$ $(O(\log M))$). נחפש את הקורס של הרצאה $prev_lecture$ בעץ הקורסים $(O(\log n))$ ובאמצעות המצביע במערך $node_ptrs_array$, ניגש למיקום $prev_lecture$ ברשימה של העץ $time \neq 0$ $(O(1))$, ושם נוסיף את ההרצאה עם הזמן המעודכן $(O(1))$.
 6. נעדכן את המצביע באינדקס ההרצאה ב- $node_ptrs_array$ לאיבר המוחזר מפונקציית ההוספה לרשימה $(O(1))$.

הסיבוכיות האפשרית הכוללת בחלוקה למקרים היא (בהתאם למקרים השונים):

$$2 \log(n) + 1 + \log(n) + 1 + 1 + \log(M) + \log(M) + \log(n) + 1 + 1 + 1 \\ = 4 \log(n) + 2 \log(M) + 6$$

או

$$2 \log(n) + 1 + \log(n) + 1 + 1 + 1 + 1 + \log(M) + \log(M) + \log(n) + 1 + 1 + 1 \\ = 4 \log(n) + 2 \log(M) + 8$$

או

$$2 \log(n) + 1 + \log(M) + 1 + 1 + \log(M) + \log(M) + \log(n) + 1 + 1 + 1 \\ = 4 \log(n) + 3 \log(M) + 6$$

מכיוון ש- $n \gg M$, סך הכל הסיבוכיות היא $O(\log(M))$. (בתרגיל נדרש $O(\log(M) + t)$, ומכיוון שהסיבוכיות הנדרשת גבוהה מהסיבוכיות שהתקבלה, סיבוכיות זו בפרט עומדת בסיבוכיות הנדרשת).

:TimeViewed

ראשית נבדוק תקינות קלט כדרוש בתרגיל. לאחר מכן נבדוק האם הקורס קיים בעץ הקורסים $O(\log(n))$. במידה והקורס קיים נבדוק חוקיות מספר ההרצאה. במקרה הצורך נחזיר שגיאה מתאימה.

נמצא את הקורס בעץ הקורסים $O(\log n)$, ונחזיר את זמן הצפייה של ההרצאה המבוקשת לפי המערך של lecture_times $O(1)$.

הסיבוכיות הכוללת היא: $2 \log(n) + 1$.

לכן סך הכל הסיבוכיות היא $O(\log(n))$.

:GetMostViewedClasses

נתחיל מהרשימה של $time \neq 0$ ומכל איבר ברשימה ניקח בהתאם את מספר הקורס ומספר ההרצאה, אותם נעדכן במערכים שאנו מקבלים. לאחר מכן, נעבור לרשימה של $time=0$ ונבצע את אותו תהליך, בהתאם למספר השיעורים המבוקשים. (אם לא הגענו למספר הנדרש נחזיר FAILURE).

כל מעבר בין הרצאות ברשימות מתבצע ב- $O(1)$, ונבצע בדיוק m איטרציות כאלה, כאשר m הוא מספר השיעורים המבוקש. הרשימות נשארות ממוינות במהלך הכנסת קורסים, הסרת קורסים וכן עדכון זמני הרצאות, ולכן פעולה זו מתבצעת בסיבוכיות של $O(m)$ ומחזירה את מספרי ההרצאות והקורסים בסדר הנדרש.

:Quit

נמחק את DS, וכך יקרא ההורס של כל מבנה הנתונים. נציב ערך NULL ב-DS. בעץ הקורסים ובעץ $time=0$ יש לכל היותר n איברים, לכל קורס יש 2 מערכים עם m איברים כל אחד (סך הכל, כל המערכים הם בעלי $2M$ איברים), בעץ $time \neq 0$ יש לכל היותר M איברים וכן סך כל האיברים בשתי הרשימות הוא M . כל הורס של מבני הנתונים מבצע פעולות כמספר האיברים שלו, ולכן הסיבוכיות הכוללת היא:

$$n + n + 2M + M + M = 2 \cdot n + 4 \cdot M$$

לכן סך הכל הסיבוכיות היא $O(n + M)$.

הוכחת סיבוכיות המקום של מבנה הנתונים:

- בעץ הקורסים `courses_tree` בכל רגע נתון יש n צמתים, כמספר הקורסים. לכל קורס שני מערכים, שגודלם הוא כמספר ההרצאות של הקורס. לכן בעץ זה יש בידינו n צמתים, m תאים במערכים של זמני ההרצאות ועוד m תאים במערכים של המצביעים להרצאות. סך הכל $n + 2m$.
- בעץ הקורסים עבור זמן צפייה אפס, `time_zero_tree`, n צמתים לכל היותר כאשר n הוא מספר הקורסים במערכת. סך הכל n .
- בעץ ההרצאות עבור זמן צפייה גדול מאפס, `time_not_zero_tree`, צמתים כמספר ההרצאות במערכת אשר להן זמן צפייה גדול מאפס. כאשר m הוא מספר ההרצאות הכולל במערכת, יהיו לנו לכל היותר m צמתים בעץ זה. סך הכל m .
- נתבונן כעת בשתי הרשימות שלנו, עבור זמן צפייה אפס ועבור זמן צפייה גדול מאפס. נשים לב כי כל הרצאה תופיע ברשימה אחת בלבד, שכן לא ייתכן כי להרצאה אחת שני זמני צפייה שונים. לכן, סך האיברים בשתי הרשימות הוא בדיוק m , מספר ההרצאות במערכת. סך הכל m .

נסכום כעת את המקום המוקצה למערכת שלנו מכלל מבני הנתונים אשר בידנו:

$$\underbrace{n + 2m}_{\text{courses_tree}} + \underbrace{n}_{\text{time zero tree}} + \underbrace{m}_{\text{time not zero tree}} + \underbrace{m}_{\text{lists}} = 2n + 4m = O(n + m)$$