

יבש של תרגיל בית רטוב 2

תיאור מבני הנתונים:

- **AVL<int, Class*> *classes_tree**
עץ AVL מדורג כאשר אנו שומרים בתור מידע נוסף את גודל תת העץ בכל חוליה. המידע בעץ זה הוא מסוג Int אבל הוא אינו משמש ולא רלוונטי, ואילו המפתח בכל עץ הוא אובייקט מסוג "שיעור" (class.h). כל אובייקט שיעור מחזיק מידע על סך הצפייה בשיעור, מזהה הקורס שלו ומזהה השיעור עצמו.
מימשנו פונקציית ההשוואה בין אובייקטים שונים של שיעור אשר ממיינת אותם ע"פ הסדר שהוגדר תחת 'GetlthWatchedClass' (סכום משך זמני הצפייה בדקות – בסדר יורד. במקרה של שוויון בסך דקות הצפייה, נעדיף את המספר המזהה של הקורס בסדר עולה, ובמקרה של שוויון נוסף נעדיף את מזהה השיעור בסדר עולה). העץ מכיל רק שיעורים שנצפו כבר.
- **HashTable<Course> *courses**
טבלת ערבול דינאמית של כל הקורסים במערכת, שמכילה אובייקטים מסוג קורס (Course) תחת (Course.h). כל אובייקט מסוג קורס מכיל גם כן טבלת ערבול דינאמית שמכילה את כל השיעורים שיש באותו קורס, מזהה הקורס ואת כמות השיעורים הנוכחית בקורס (שמאותחלת ל 0 ועולה ב 1 עם כל הוספת שיעור חדש לקורס).
אבחנה חשובה בהתאם להגדרות התרגיל הרטוב: מכיוון שאין פונקציית הסרה של שיעור, אם לקורס מסוים יש x שיעורים אז בהכרח מזהי השיעורים של הקורס יהיו מ 0 עד x-1!
טבלת הערבול מאותחלת עם גודל קבוע מראש וגדלה/קטנה בהתאם לעומס בטבלה. טבלת הערבול של הקורסים מערבלת ע"פ מזהי הקורס ולכל קורס – טבלת הערבול של השיעורים מערבלת לפי מזהה השיעור (שימוש בפונקציית מודולו שנלמדה בתרגול) וממושת בשיטת double hasing שאינה צריכה לדעת את סדר גודל הקלט כי היא משנה את גודלה דינמית כפי שלמדנו בתרגול.

סיבוכיות המקום של המבנה:

נניח כי יש n קורסים ו-m שיעורים במערכת. אזי יהיו לנו n קורסים בטבלת הערבול של הקורסים וסה"כ m שיעורים בטבלת הערבול של הקורסים ופרוסים בטבלת השיעורים של כל קורס בהתאם לשיוכם לקורס. במקרה הגרוע, היו צפיות בכל השיעורים ולכן יהיו m שיעורים בעץ השיעורים הכללי (כמפורט לעיל).

סה"כ סיבוכיות מקום: $O(n + m + m) = O(n + m)$ כנדרש.

תיאור מימוש הפונקציות בתרגיל:

נציין כי בכניסה לכל פונקציה אנחנו בודקים את תקינות הקלט בהתאם לפונקציה הנוכחית ומחזירים שגיאה בהתאם – כל זה $O(1)$.

void * Init()

- אתחול עץ ריק – $O(1)$ במקרה הגרוע.
- אתחול טבלת ערבול ריקה – $O(1)$ במקרה הגרוע.
- סה"כ סיבוכיות: $O(1)$ במקרה הגרוע.

StatusType AddCourse(void* DS, int courseID)

- מחפשים בטבלת הערבול של הקורסים אם הקורס כבר קיים – $O(1)$ בממוצע על הקלט באופן משוערך, אם קיים כבר אז מחזירים שגיאה.
- אחרת, יוצרים אובייקט קורס ב – $O(1)$ במקרה הגרוע, ומכניסים אותו לטבלת הקורסים ב – $O(1)$ בממוצע על הקלט באופן משוערך.
- סה"כ סיבוכיות: $O(1)$ בממוצע על הקלט באופן משוערך.

StatusType RemoveCourse(void* DS, int courseID)

- מחפשים את הקורס בטבלת הקורסים ב - $O(1)$ בממוצע על הקלט באופן משוערך, אם לא קיים נחזיר שגיאה.
- אחרת, במקרה הגרוע, נעבור על כל m ההרצאות של הקורס ונסירם אחד אחד מעץ השיעורים הכללי. בהכרח קיימים שיעורים עם המזהים הללו ע"פ ההבחנה לעיל, אך לא בהכרח קיימים בעץ כי אולי הם טרם נצפו. במקרה הגרוע, כולם נצפו ולכן קיימים בעץ אז נבצע m פעולות הסרה מעץ השיעורים, כאשר לכל שיעור נחפש בטבלת השיעורים בקורס את זמן הצפיה של השיעור ב - $O(1)$ בממוצע על הקלט באופן משוערך – במקרה הגרוע סה"כ סיבוכיות $O(m \log M)$ בממוצע על הקלט באופן משוערך, לפי הנלמד בהרצאה ובתרגול, כש M הוא סך השיעורים במערכת.
- לכל קורס נמחק את כל השיעורים שלו מטבלת הערבול ב - $O(m)$ במקרה הגרוע.
- לבסוף נסיר את הקורס מטבלת הקורסים ב - $O(1)$ בממוצע על הקלט באופן משוערך.
- סה"כ סיבוכיות: $O(m \log M) = O(m \log M)^* + O(m) + O(1)^* + O(1)^*$ **בממוצע על הקלט באופן משוערך.**
- * (בממוצע על הקלט באופן משוערך).

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

- מחפשים את הקורס בטבלת הקורסים ב - $O(1)$ בממוצע על הקלט באופן משוערך, אם לא קיים נחזיר שגיאה.
- אחרת, ניצור אובייקט שיעור חדש ב - $O(1)$ עם מזהה הקורס, מזהה השיעור שיהיה כמות השיעורים של הקורס הנוכחי (זה יהיה מזהה השיעור ע"פ ההבחנה בתחילת התרגיל) וזמן צפייה כולל של 0.
- נעדכן את המצביע למזהה השיעור החדש ב - $O(1)$.
- נכניס את השיעור בחדש לטבלת הערבול של השיעורים של הקורס ב - $O(1)$ בממוצע על הקלט באופן משוערך.
- סה"כ סיבוכיות: $O(1)$ בממוצע על הקלט באופן משוערך.

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

- מחפשים את הקורס בטבלת הקורסים ב - $O(1)$ בממוצע על הקלט באופן משוערך, אם לא קיים נחזיר שגיאה.
- אחרת, אם מתקיים שפרמטר מזהה השיעור ועוד 1 גדול מכמות השיעורים של הקורס אז נחזיר שגיאה ב - $O(1)$.
- אחרת, נחפש בטבלת השיעורים בקורס את השיעור ב - $O(1)$ בממוצע על הקלט באופן משוערך.
- במקרה הגרוע, אם השיעור קיים בעץ – נמחק את השיעור שמצאנו מהעץ ב - $O(\log M)$.
- נעדכן את זמן הצפייה בשיעור ב - $O(1)$ (מתעדכן גם בטבלת הערבול של הקורס).
- ניצור אובייקט שיעור חדש עם זמן מעודכן, מזהה הקורס ומזהה השיעור המתאימים ב - $O(1)$ ונכניסו חזרה לעץ ב - $O(\log M)$.
- סה"כ סיבוכיות: $O(\log M + 2) = O(\log M) + O(1) \cdot 3 + O(\log M) \cdot 2 + O(1)^* \cdot 2$ **בממוצע על הקלט.**

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

- מחפשים את הקורס בטבלת הקורסים ב - $O(1)$ בממוצע על הקלט באופן משוערך, אם לא קיים נחזיר שגיאה.
- אחרת, אם מתקיים שפרמטר מזהה השיעור ועוד 1 גדול מכמות השיעורים של הקורס אז נחזיר שגיאה ב - $O(1)$ במקרה הגרוע.

- אחרת, נחפש את השיעור בטבלת הערבול של השיעורים בקורס ב - $O(1)$ בממוצע על הקלט באופן משוערך (בהכרח קיים השיעור ע"פ ההבחנה בתחילת הקובץ) ונבצע השמה של זמן הצפייה שלו לפרמטר המצביע.
- סה"כ סיבוכיות: $O(1)$ בממוצע על הקלט.

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

- נשים לב שע"פ הגדרת עץ השיעורים, גודל העץ הוא כמות השיעורים הנצפים במערכת, לכן נבדוק האם הוא קטן מ - i ואם כן נחזיר שגיאה ב - $O(1)$ במקרה הגרוע.
- נבצע **Select(i)** על העץ על מנת לקבל את השיעור המתאים וממנו נקבל המזהה שלו ושל הקורס שלו ב - $O(\log M)$ במקרה הגרוע. נשים לב כי מאופן הגדרת הסדר בין צמתי העץ (פונקציית ההשוואה בין השיעורים - שפירטנו בהתחלה) ובשל העובדה כי עץ השיעורים הוא עץ דרגות עם מידע נוסף על הצמתים של גודל תת העץ - בהכרח נקבל את השיעור שרצינו כמפורט בתרגיל.
- סה"כ סיבוכיות: $O(1) + O(\log M) = O(\log M) = O(\log M + 2)$ במקרה הגרוע.

`void Quit(void **DS)`

- נבצע הריסה ומחיקה לגמרי של עץ השיעורים - $O(m)$ במקרה הגרוע.
- נבצע הריסה לגמרי של טבלת הערבול של הקורסים כאשר לכל קורס נמחק את טבלת השיעורים שיש אצלו ואז נמחק את חוליית הקורס מהטבלת הקורסים - סיבוכיות $O(m + n)$ במקרה הגרוע.
- סה"כ סיבוכיות: $O(m + n) + O(m) = O(m + n)$ במקרה הגרוע.