

(812432) שם הקורס מבני נתונים | תרגיל רטוב 2

שם: ניב יעל והוד בדיחי | ת"ז: 315314922, 313349185

10 בינואר 2021

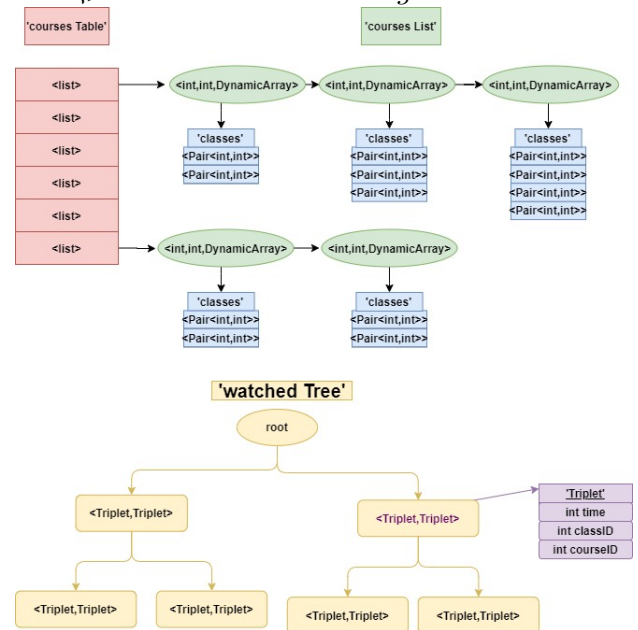
דיאגרמת מבנה הנתונים 'CourseManager'

רעיון כללי

נחזיק טבלת ערבול דינאמית שתהיה אחראית על אחזקת כלל הקורסים במערכת, בתוך הטבלה כל קורס יכיל מערך דינאמי שיתעדכן בהתאם לכמות ומספר הצפיות בשיעוריו.
נחזיק עץ דרגות מאוזן שיכיל באופן ממין כנדרש בסוף כל פעולה את זמני הצפייה של כל שיעור במערכת שנצפה.

תיאור המבנה

מבנה הנתונים 'CoursesManager' בנוי מעץ וטבלת ערבול:



עיקרי המבנה:

- 'watched_tree' - עץ דרגות, מחזיק בצומת מפתח וערך מסוג 'Tripletkey' (מפורט ב"טיפוסי נתונים נוספים"). יחס הסדר המוגדר על העץ הוא $time \rightarrow courseID \rightarrow courseNum$ כאשר הצומת בעלת הזמן הכי גדול היא השמאלית ביותר (הפוך מעץ חיפוש רגיל) והמיון המשני והשלישוני ע"י השוואת int רגילה, העץ מחזיק את כלל שיעורי הקורסים במערכת שנצפו
- 'courses_table' - טבלת ערבול שמטפלת בהתנגשויות בצורת chain_addressing ופונקציית הערבול שלה היא מודלו גודל הטבלה הנוכחי.
- הרשימה המקושרת מנוהלת ע"י מחלקת 'course_list' (מפורט ב"טיפוסי נתונים נוספים")

איבר של הרשימה המקושרת בכל תא ערבול מורכב משלושה שדות : שם הקורס, מספר השיעורים שאותחלו עד כה , מערך `Dynamic_Array` (מפורט ב"טיפוסי נתונים נוספים") שמכיל את מספר הצפיות וזמן הצפייה של כל שיעור.

טיפוסי נתונים נוספים:

- `TripletKey` - טיפוס נתונים בעל שלוש שדות

(1 `int time` מחזיק את הזמן

(2 `int courseID` מחזיק את מספר הקורס

(3 `int classNum` מחזיק את מספר השיעור

על המפתח מוגדר יחס סדר ראשון על ידי `time` (שעובד הפוך מאופרטור השוואה "`>`" רגיל בין `int`) משני על ידי `courseID` ושלישון ע"י `classNum` שעובדים עם אופרטור "`>`" זהה לשל `int` תפקידו שמירה על יחס סדר בעץ `'watched_tree'` ואפסון מידע.

- `Dynamic_array` - טיפוס שגודלו משתנה בהתאם למספר האיברים שהוכנסו

המערך עובד באופן זהה למערך שראינו בתרגול רק בלי מחיקות כלל , כאשר בהכנסת שיעור חדש לקורס נשתמש בפונקציית `push()` שמכניסה את האיבר הבא בצורה דומה לשל מחסנית (בצורה זו הבטיח שאנחנו מכניסים כנדרש ב `addClass`)

תפקידו אחסון זמני הצפייה ומספר הצפיות של השיעורים שמתאימים לכל קורס במבנה

- `'course_list'` - מחלקה שמורכבת מרשימה מקושרת ,

ברשימה המקושרת מבנה נתונים של `'Triplet'` שמרכיב 3 שדות :

(1 `int courseID`

(2 `int numOfClass`

(3 `DynamicArray()`

תפקיד הרשימה המקושרת לטפל בהוספת \ הוצאת מידע בהכנסת \ הוצאה של קורסים ושיעורים חדשים ובהוספה \ הוצאת זמני צפייה .

מימוש הפונקציות וחישוב סיבוכיות הזמן

אבחנה:

פאקטור העומס α הוא היחס בין m, n (מספר האיברים בטבלה וגודל הטבלה).

אנו מניחים כי פונקציית המודלו מקיימת את הנחת הפיזור האחד כי מודלו גודל הטבלה ממפה איבר אקראי בהסתברות של $\frac{1}{m}$ עבור כל תא.

בנוסף היא על , וניתן לחשבה ב $O(1)$ כנדרש.

טבלת הערובול דינאמית כך שאם היא מלאה גודלה יוכפל בשתיים , ואם היא רבע מלאה גודלה ייקטן בשתיים לכן מתקיים תמיד כי

$$\frac{m}{4} \leq n \leq m$$

(מלבד באתחול הראשוני)

בדרך זו נבטיח כי הגדלה והקטנה של הטבלה הדינאמית תהיה ב $O(1)$ משוערך כפי שראינו בתרגול על מערכים דינאמיים ונבצע העתקה של מצביעי הרשימות המקושרות.

בשל היחס הקבוע בין m, n נבטיח שיתקיים בממוצע על הקלט משוערך

$$\frac{m}{n} = \underbrace{O(1)}_{\text{amortized, avg}} \quad \checkmark$$

וכל פעולת הוצאה \ הכנסה \ סריקת הטבלה הדינאמית תהיה ב $O(1)$ משוערך בממוצע על הקלט.

`void * Init()`

תחילה מתבצעת הקצאה דינמית של *CoursesManager* ומופעל הבנאי ברירת המחדל שלו.

עץ 'watched_tree' מאותחל ב $O(1)$ ללא צמתים כשהשורש שלו מצביע ל *nullptr*

טבלת ערובול בגודל התחלתי 8 'courses_table' עם תאים ריקים מאותחלת ב $O(1)$.

סך הכל $O(1)$ ✓ סיבוכיות זמן.

`StatusType AddCourse(void* DS, int courseID)`

תחילה נבדוק שערכי הקלט תקינים במידה ולא נחזיר שגיאה.

נבדוק האם הקורס קיים אם כן נחזיר שגיאה אחרת נמשיך, נבצע זאת באמצעות פונקציית ה *hash* ונרוץ על הרשימה 'course_list' כל זה מבוצע ב $O(1)$ בממוצע על הקלט(כפי שראינו בתרגול ומהאבחנה לעיל).

נכניס את מספר הקורס למקום המתאים באמצעות *hash()* לנתיב למקום המתאים ב 'course_list' ב $O(1) = O(\alpha)$. בממוצע על הקלט משוערך .

נאתחל את שדותיו(מספר קורס *courseID*, מונה השיעורים ל 0 , ומערך דינאמי בגודל 8 ריק) ב $O(1)$.
סה"כ

$$\underbrace{O(1)}_{\text{avg}} + \underbrace{O(1)}_{\text{avg, amortized}} + O(1) = \underbrace{O(1)}_{\text{avg, amortized}} \quad \checkmark$$

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

תחילה נבדוק שערכי הקלט תקינים במידה ולא נחזיר שגיאה.

נבדוק האם הקורס קיים , אם לא נחזיר שגיאה אחרת נמשיך . נבצע זאת באמצעות פונקציית ה *hash* ונרוץ על הרשימה 'course_list' כל זה מבוצע ב $O(1)$ בממוצע על הקלט .

ניגש לתא המתאים ברשימת 'course_list' של הקורס, ונעבור על המערך בשדה המערך הדינמי של *courseID*.

בעזרת הערכים השמורים במערך (זמן , מספר קורס , מספר השיעור) נמחק את כל m שיעוריו של $courseID$ מ $'watched_tree'$ זהו עץ מאוזן וכל מחיקה תבוצע בו ב $O(\log(M))$ סה"כ $O(m \cdot \log(M))$.
 כעת נמחק את המערך הדינאמי שגודלו לכל היותר $2m$ (במקרה שהוא בדיוק סיים הגדלה) לכן מחיקה תעלה לנו $O(m)$.
 לאחר מכן הסרת האיבר מטבלת הערבול יהיה ב $O(1)$ משוערך, בממוצע על הקלט.
 סה"כ

$$\underbrace{O(1)}_{ammortized} + O(m \cdot \log(M)) + O(m) + \underbrace{O(1)}_{ammortized, avg} = \underbrace{O(m \cdot \log(M))}_{ammortized, avg} \checkmark$$

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

תחילה נבדוק שערכי הקלט תקינים במידה ולא נחזיר שגיאה.

ניגש לאיבר המתאים בעזרת *hash* על *courseID* וריצה על 'course_list' מתאים ב $O(1)$ ממוצע על הקלט.

אם הוא לא קיים נחזיר שגיאה, אם קיים ניגש לשדה המערך הדינאמי ונבצע פעולת *Push()* כדי להוסיף שיעור חדש לקורס ללא צפיות וללא זמן צפייה כולל.

בעזרת מונה השיעורים בשדה הנוסף נדע איזה מספר שיעורי עלינו להוסיף, בסיום נעדכן את המונה ונגדילו ב 1, סה"כ $O(1)$ משוערך.

סה"כ

$$\underbrace{O(1)}_{avg} + \underbrace{O(1)}_{ammortized} = \underbrace{O(1)}_{ammortized, avg}$$

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

ניגש לאיבר המתאים בעזרת *hash* על *courseID* וריצה על 'course_list' מתאים ב $O(1)$ סה"כ $O(1)$ בממוצע על הקלט.

אם הוא לא קיים או ששדה מספר השיעורים בקורס הכולל קטן מ *classID* נחזיר שגיאה.

אם קיים ניגש לשדה המערך הדינאמי ובאינדקס ה *classID* ונוסיף לזמן הקיים שם את *time*, ולשדה מספרי הצפיות 1 כל זה מבוצע ב $O(1)$.

נמחק את האיבר מהעץ 'watched_tree' שהחזיק את זמן הצפייה הישן של השיעור ב $O(\log(M))$ ונוסיף את האיבר לעץ עם הזמן המעודכן ב $O(\log(M))$

סה"כ

$$O(1) + \underbrace{O(1)}_{avg} + O(\log(M)) = \underbrace{O(\log(M+2))}_{avg} \checkmark$$

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

תחילה נבדוק שערכי הקלט תקינים במידה ולא נחזיר שגיאה.

ניגש לאיבר המתאים בעזרת *hash* על *courseID* וריצה על 'course_list' מתאים ב $O(1)$ סה"כ $O(1)$ בממוצע על הקלט.

אם הוא לא קיים או ששדה מספר השיעורים בקורס הכולל קטן מ *classID* נחזיר שגיאה.

אם קיים ניגש לשדה המערך הדינאמי ונוציא ממנו את ערך זמן הצפייה הכולל של השיעור באינדקס ה *classID* ב $O(1)$ ל *timeViewed*

סה"כ

$$\underbrace{O(1)}_{avg} + O(1) = \underbrace{O(1)}_{avg} \checkmark$$

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

תחילה נבדוק שערכי הקלט תקינים במידה ולא נחזיר שגיאה.

ניגש לעץ 'watched_tree' שמחזיק רק את השיעורים שנצפו, אם מספר איבריו קטן מ i נחזיר שגיאה. לאחר מכן בגלל ש 'watched_tree' הוא עץ דרגות נוכל להשתמש באלגוריתם מההרצאה שראינו ל rank_tree למציאת האינדקס ה i ב $O(h) = O(\log(M))$. בעזרת טיפוס הנתונים של הצומת 'TripletKey', ויחס הסדר המוגדר על העץ (כפי שמוסבר ב "טיפוסי נתונים נוספים") נקבל את האיבר ה- i לפי דרישות המיון של התרגיל נאפסן את ערכיו במצביעים שקיבלנו ונסיים את הפעולה
סה"כ

$$O(\log(M)) \checkmark$$

כנדרש.
void Quit(void **DS)

הריסת המבנה 'CoursesManager' תפעיל שרשרת הורסים תחילה יפעל ההורס של העץ 'watched_tree' בעזרת סיור inorder נהרוס את כל צמתיו סך כל הסיבוכיות היא $O(m)$. כעת יפעל ההורס של טבלת הערבול 'courses_table' נעבור על כל הרשימות המקושרות שבתאי הטבלה, סך הכל יש $O(n)$ איברים בסך הרשימות (כמספר הקורסים). כל איבר כזה יפעיל הורס שיהרוס שני שדות ב $O(1)$ ואת שדה המערך הדינמי שלו. מספר האיברים במערך הדינמי, מאופן הגדלתו, הוא לכל היותר $2 \cdot m_l$ כאשר m_l מספר השיעורים של הקורס l נקבל כי כלל הריסות המערכים יקיימו

$$\sum O(2m_l) = O(m)$$

הריסת השדות הנוספים יבוצעו ב $O(1)$ לכל אחד מהאיברים. ולבסוף הטבלה עצמה מאופן בנייתה יש לכל היותר $4n$ תאים ולכן תהרס ב $O(4n) = O(n)$ סה"כ

$$O(m) + O(n) + O(m) + O(4n) = O(m + n) \checkmark$$

חישוב סיבוכיות המקום

- טבלת הערבול תחזיק מאופן הגדלתה לכל היותר $4n$ תאים לכן סה"כ $O(n)$
- הרשימות המקושרות בטבלת הערבול מחזיקות יחדיו את סך מספר הקורסים יש n כאלה ולכן $O(n)$
- המערכים הדינאמיים ברשימות המקושרות מאופן הגדלתם מחזיקים לכל היותר $2m_l$ תאים כל אחד, כאשר m_l מספר השיעורים של הקורס, אם נסכום את כולם נקבל

$$\sum 2m_l = 2m = O(m)$$

• בעץ יש את השיעורים שנצפו לכן יש שם לכל היותר $O(m)$ צמתים

סה"כ נקבל

$$O(n) + O(n) + O(m) + O(m) = O(m + n)$$