

מבני נתונים - תרגיל רטוב 2

17 בינואר 2021

מבני הנתונים וטיפוסים בהם השתמשנו

$AVL_Tree < KEY, VAL >$ - עץ דרגות מאוזן המבוסס על עץ AVL :

עץ Avl גנרי דו כיווני (הורה מצביע לילד וילד מצביע להורה), כפי שנלמד בהרצאה. הצמתים ממוינים לפי ערך המפתח. כל צומת מכיל מפתח, ערך, ואת מספר הצמתים בתת העץ ששורשו הוא אותו צומת.

הפעולות המוגדרות וסיבוכיות הזמן והמקום של המבנה כפי שנלמדו בכיתה. (גם הפעולה $select(k)$ נעשית בסיבוכיות זמן ומקום $O(\log n)$, בדומה לחיפוש בעץ מאוזן. במהלך החיפוש של הצומת בעלת הדרגה k אנו עוברים בכל רמה בעץ פעם אחת לכל היותר ולכן $O(h)$. וכיוון שהעץ הוא מאוזן החיפוש נעשה ב $O(\log n)$).

$List < T >$ - רשימה מקושרת דו כיוונית:

אברי הרשימה הם טיפוס $ListNode < T >$, המכיל פוינטר חכם ל ערך T , מצביע לאיבר לפני ברשימה ומצביע לאיבר אחרי ברשימה. הפעולות המוגדרות וסיבוכיות הזמן והמקום של המבנה כפי שנלמדו בכיתה.

$DynamicArray < T >$ - מערך דינמי:

מערך התומך בהוספה למיקום הראשון הפנוי והגדלת המערך פי שתיים לאחר הוספה למיקום האחרון הפנוי. הפעולות המוגדרות וסיבוכיות הזמן והמקום של המבנה כפי שנלמדו בכיתה.

$HashTable < T >$ - טבל ערבול דינמית:

טבלת ערבול בשיטת $chain hashing$ הבנויה על עיקרון מערך דינמי. הטבלה מחזיקה מערך של רשימות לטיפוס המאוחסן. בהוספת אובייקט לטבלה, אם ההוספה מגדילה את פקטור העומס ליותר מ2, מערך הרשימות גדל פי 2 והאובייקטים עוברים ערבול מחדש. בהסרת אובייקט מהטבלה, אם ההסרה מקטינה את פקטור העומס לפחות מ0.5, מערך הרשימות קטן פי 2 והאובייקטים עוברים ערבול מחדש. פונקציית הערבול הנבחרת היא מציאת מודולו גודל הטבלה: $id \% size$ כאשר $size$ הוא גודל הטבלה. שאר הפעולות, וסיבוכיות הזמן והמקום של המבנה, כפי שנלמדו בכיתה.

Lecture - אובייקט המייצג הרצאה:

שדות:

courseID - מספר טבעי חיובי (*int*) המייצג את מזהה הקורס שאליו שייכת ההרצאה.
lectureID - מספר טבעי אי שלילי (*int*) המייצג את מזהה ההרצאה.
num_views - מספר טבעי אי שלילי המייצג כמה צפיות יש להרצאה.
לאובייקט הנ"ל מוגדר יחס סדר בעזרת אופרטורי השוואה (גדול, קטן, שווה) המוגדר באופן הבא:

- (1) לפי *num_views* - מספר צפיות בסדר ירוד (הרצאה l_1 בעלת מספר צפיות גדול יותר מאלה של l_2 תקיים $l_1 > l_2$).
אם מתקיים שוויון במספר הצפיות:
- (2) לפי *courseID* - מזהה הקורס בסדר עולה (בהינתן שוויון במספר הצפיות, הרצאה l_1 בעלת מזהה קורס קטן מזה של l_2 תקיים $l_1 > l_2$).
אם מתקיים שוויון גם במזהה הקורס:
- (3) לפי *lectureID* - מזהה ההרצאה בסדר עולה.
במידה ומתקיים שוויון בכל שלושת השדות, נגיד ש l_1 שווה ל l_2 .

פעולות מוגדרות:

אתחול: אתחול שדות המבנה.
השמת מספר קבוע של ערכים מסוג *int*.
הוספת זמן צפיה: עדכון שדה זמן הצפיה לפי פרמטר.
עדכון שדה מסוג *int*.
שליפת זמן צפיה: החזרת שדה זמן הצפיה.
קריאה של שדה מסוג *int*.
אופרטורי השוואה: השוואה בין הרצאות כפי שתוארה לעיל.
השוואה של *int* ימים.

סיבוכיות זמן של כלל הפעולות:

כל הפעולות הן בסיבוכיות זמן ומקום $O(1)$ (פעולות על מספר קבוע של *int* ימים).

סיבוכיות מקום:

$O(1)$, השדות של *Lecture* הן מספר קבוע של קבועים מסוג *int*.
הפעולות לא מקצות זכרון חדש ולכן גם הן $O(1)$ כל אחת.

Course - אובייקט המייצג קורס:

מטרה:

מטרת המבנה היא להכיל את המידע הרלוונטי לקורס מסוים והרצאותיו.

שדות:

courseID - מספר טבעי חיובי (*int*) המייצג את מזהה הקורס.
lecture_arr - מערך דינמי של *Lecture*, המאפשר גישה בזמן קבוע לכל ההרצאות של הקורס.

פעולות מוגדרות:

אתחול (מפרמטרים או כהעתקה עמוקה):
אתחול השדות. אתחול מערך דינמי ריק הוא ב $O(1)$ ולכן גם אתחול קורס הוא ב $O(1)$.
אופרטור השמה:
העתקה עמוקה של המערך הדינמי, מתבצעת ב $O(m)$ כאשר m הוא מספר ההרצאות בקורס בעת ההשמה.

שליפת שדות: (Getters):

נעשה ב $O(1)$, כיוון שבשליפת במערך הדינמי מועבר רפרנס.

הריסה:

הריסת המערך הדינמי – מעבר על מספר ההרצאות בקורס והריסתן אחת אחת ב- $O(m)$.

סיבוכיות מקום:

המערך הדינמי שומר m הרצאות. מעבר לכך, כל השדות שלו ושל הקורס הם טיפוסים פשוטים. מכאן שסיבוכיות המקום של $Course$ היא $O(m)$.

נציין כי הפונקציות של הפעולות לא רקורסיביות ולא מקצות זכרון חדש ולכן סיבוכיות המקום שלהן היא $O(1)$.

אובייקט Boom ולו השדות הבאים:

$courses$ - טבלת ערבול של קורסים.

$lectures$ - עץ דרגות של $Lecture$, ממיון לפי אופרטורי ההשוואה של מחלקת $Lecture$. כלומר, מספר צפיות בסדר יורד, לאחר מכן מזהה קורס בסדר עולה, ולאחר מכן מזהה הרצאה בסדר עולה.

עץ זה מכיל רק את ההרצאות שלהן יותר מ 0 צפיות.

פעולות מוגדרות:

אתחול - נקרא על ידי הפונקציה $Init$ בממשק התרגיל:

אתחול של טבלת הערבול ושל עץ הדרגות. אתחול של כל אחד מהמבנים הללו הוא $O(1)$ ולכן אתחול המבנה הוא $O(1)$.

- $AddCourse$

מבצעת הוספה של קורס חדש לטבלה $courses$.

כנלמד בהרצאות, הכנסה לטבלת ערבול היא $O(1)$ בממוצע על הקלט, משוערך.

- $RemoveCourse$

מבצעת הסרה של קורס מהטבלה $courses$, ושל כל ההרצאות בעץ $lectures$ ששייכות לקורס הזה ובעלות צפיות. הסרה של הרצאה מהעץ היא בסיבוכיות זמן $O(\log M)$ כאשר M הוא מספר ההרצאות על פני כל הקורסים בעת ההוספה.

עבור לא יותר מ m הרצאות שמוסרות מהעץ, כאשר m הוא מספר ההרצאות בקורס המוסר, נקבל כי סיבוכיות הזמן של הסרת כל ההרצאות היא $O(m \log M)$.

בנוסף, מתבצעת הסרה של הקורס מטבלת הערבול, וסיבוכיות הזמן לכך היא $O(1)$ בממוצע על הקלט, משוערך.

לכן סיבוכיות הזמן של הפעולה היא $O(m \log M)$ בממוצע על הקלט, משוערך.

- $AddClass$

מוצאת את הקורס הרלבנטי לו רוצים להוסיף הרצאה.

שליפה מטבלת ערבול היא ב- $O(1)$ בממוצע על הקלט.

הוספת ההרצאה היא הוספת איבר חדש למערך הדינמי של ההרצאות בקורס שמצאנו.

אתחול של הרצאה נעשה ב- $O(1)$.

הכנסה למערך דינמי מתבצעת ב- $O(1)$ בממוצע על הקלט משוערך.

לכן סה"כ סיבוכיות הזמן של $AddClass$ היא $O(1)$ בממוצע על הקלט משוערך.

- $WatchClass$

מוסיף זמן צפייה להרצאה מסויימת. עדכון מיקום ההרצאה בעץ $lectures$ מתבצע על ידי

הסרת ההרצאה אם היא כבר בעץ, הוספת הזמן הנדרש, והוספת ההרצאה מחדש לעץ עם הזמן המעודכן.

הוצאה והכנסה מעץ דרגות המבוסס על עץ AVL מתבצעות ב- $O(\log M)$.

גם כאשר $M = 0$, מתבצע מספר קבוע של פעולות מציאת הקורס הרלבנטי, ולכן על מנת למנוע מהביטוי להתאפס ניתן לכתוב בסך הכל כי הפעולה מתבצעת בסיבוכיות זמן של $O(\log(M + 2))$.

- TimeViewed

מוצאת את הקורס הרלבנטי בו נמצאת ההרצאה המבוקשת.
 שליפה מטבלת ערבול היא ב $O(1)$ בממוצע על הקלט.
 מציאת ההרצאה במערך הדינמי של הקורס המאחסן את ההרצאות היא $O(1)$ כיוון ש $classID$ הוא גם האינדקס של ההרצאה במערך.
 סך הכל הפעולה מתבצעת ב $O(1)$ בממוצע על הקלט.
 -GetIthWatchedClass
 מבצעת פעולת $select$ על עץ הדרגות כפי שנלמד בתרגול, ב $O(\log(M))$.
 פעולת $select$ מוצאת את האיבר i הכי קטן, ולכן כדי למצוא את האיבר i הכי גדול נחפש באמצעות $select$ את האיבר $k - i + 1$ הכי קטן, כאשר k הוא מספר האיברים בעץ.
 כמו ב $WatchClass$ מתקיימות פעולות בזמן קבוע גם אם העץ ריק, ולכן ניתן סך הכל לכתוב כי סיבוכיות הזמן היא $O(\log(M + 2))$.
 - Quit

שחרור הזכרון מהעץ ומהטבלה ואיפוס הפוינטרים.
 שחרור עץ נעשה באמצעות סיור $PostOrder$ רקורסיבי על העצים.
 עץ ההרצאות משוחרר ב $O(m)$ כש m מספר ההרצאות, כי כל איבר בעץ משתחרר ב $O(1)$.
 טבלת הקורסים משתחררת ב $O(n)$, נוכיח זאת:
 • אם $n \geq m$: נסמן את מספר ההרצאות בקורס i ב m_i . מתקיים: $\sum_{i=1}^m m_i = m$, שכן סך ההרצאות בכל הקורסים הוא סך ההרצאות.
 שחרור צומת של הקורס i בעץ הקורסים היא $O(m_i)$ - שחרור של קורס כולל שחרור המערך הדינמי של ההרצאות שלו. כל הרצאה משתחררת ב $O(1)$, ולכן סך הכל קורס משתחרר ב $O(m_i)$.
 לכן שחרור סך הצמתים הוא:

$$\sum_{i=1}^n O(n_i) = O\left(\sum_{i=1}^n n_i\right) = O(n)$$

קבלנו אזי ששחרור מבנה הנתונים הוא $O(n) + O(n) = O(n)$ ולכן גם $O(n + m)$ כיוון שידוע ש $n \geq m$.
 • אם $m > n$, שחרור ההרצאות בכל קורס "נבלע" על ידי המעבר על כל אחד מהקורסים. לכן כולל שחרור n ההרצאות נקבל כי הפעולה מתבצעת ב $O(m + n)$.

סיבוכיות מקום:

המבנה מנהל זכרון של עץ דרגות עבור ההרצאות וטבלת ערבול עבור הקורסים.
 הזכרון הדרוש עבור עץ ההרצאות הוא $O(m)$ כאשר m הוא מספר ההרצאות הכולל.
 הזכרון הדרוש עבור טבלת הקורסים הוא, משיקולים זהים לשיקולי סיבוכיות השחרור של העץ שצוינו לעיל, $O(m + n)$.
 $(\sum_{i=1}^n O(n_i) = O(\sum_{i=1}^n n_i) = O(n))$
 לכן סך הכל סיבוכיות המקום של המבנה היא $O(m + n)$.
 נעיר שחיפוש בעץ דרגות מבוסס AVL , בין היתר לצורכי הוספה והסרה של איברים, ממומש אצלנו באמצעות רקורסיה.
 הפונקציה נקראת עבור כל איבר במסלול החיפוש של האיבר המבוקש. במקרה הגרוע האיבר המבוקש לא נמצא בעץ ולכן עומק הרקורסיה יהיה כעומק העץ. כידוע עץ AVL מאוזן ולכן עומק העץ הוא $O(\log(n))$, כלומר מחסנית הקריאות לא מוסיפה לסיבוכיות המקום.