**גיליון רטוב מספר 2 – חלק יבש  
  
מגישים:   
נועם יעקבי - 316600782   
עידו קולמן - 316465202   
  
תיאור מבנה הנתונים הראשי שלנו - CourseManager:**  
מבנה הנתונים הראשי שלנו נקרא CourseManager ומטרתו אחסון מידע על קורסים, השיעורים ששייכים לכל קורס וזמני הצפייה של כל שיעור.  
מבנה זה מכיל טבלת ערבול דינאמית, ועץ דרגות על בסיס AVL.  
מבנה הנתונים שלנו שומר בטבלת הערבול הדינאמית מסוג Chain Hashing את כל הקורסים שמוכנסים למערכת, כאשר כל קורס שומר מערך דינאמי המכיל את כל השיעורים שמועברים בו.  
כמו כן אנו שומרים את מספר השיעורים במערכת וכן שומרים עץ דרגות על בסיס AVL עבור כל השיעורים במערכת שיש להם זמן צפייה גדול מ-0.  
  
**הסבר על מבני הנתונים שמבנה הנתונים הראשי שלנו מכיל:**  
  
**טבלת ערבול דינאמית מסוג Chain Hashing בשם** **courses:**   
מכיל את כל הקורסים שיש במערכת.   
לפני הכנסה לטבלה נפעיל על המזהה של הקורס את פונקציית הערבול שלנו,  
כאשר בעת הכנסה (בתא הנכון לפי הערך המוחזר מפונקציית הערבול) אם התא ריק נכניס את הקורס למקום זה בטבלה, אחרת נשרשר אותו בסוף הקורסים שקיימים כבר בתא זה בטבלה.  
עבור הטבלה אנו שומרים את כמות הקורסים הנמצאים בה בכל עת ואת כמות הקורסים המקסימלית שהיא יכולה להכיל.   
על מנת לשמור על יעילות, הטבלה מגדילה ומקטינה עצמה לפי כמות האיברים בטבלה - היא מגדילה עצמה פי 2 כאשר ממנה מלאה, ומקטינה עצמה פי 2 כאשר ממנה מלאה.  
  
כל איבר בטבלה מכיל:

* ***id*** - המזהה של הקורס.
* ***root***– מצביע לתחילת המערך הדינאמי של השיעורים השייכים לקורס זה, המזהה של כל שיעור השמור בטבלה זהה למיקומו בטבלה.
* ***count*** – כמות השיעורים השייכים לקורס.
* ***capacity*** – כמות השיעורים המקסימלית שהקורס יכול לשמור בטבלת השיעורים הדינאמית שלו.
* ***next*** – מצביע לאיבר הבא ברשימה המקושרת של טבלת הערבול.
* ***prev*** - מצביע לאיבר הקודם ברשימה המקושרת של טבלת הערבול.

**עץ דרגות על בסיס AVLבשם** **classes:**   
מכיל את השיעורים במערכת שיש להם זמן צפייה גדול מ-0.  
כל השיעורים ממוינים קודם לפי זמן הצפייה, לאחר מכן לפי המזהה של הקורס ולאחר מכן לפי המזהה של השיעור.  
כל צומת מכיל:

* ***class\_id*** – המזהה של השיעור.
* ***course\_id*** – המזהה של הקורס אליו השיעור שייך.
* ***time*** – זמן הצפייה של השיעור.
* ***rank*** – כמות הבנים של הצומת ועוד הוא עצמו, נשמר על מנת לאפשר מציאה יעילה של השיעור ה-i בעץ.
* ***height*** – גובה הצומת בעץ, נשמר על מנת לאפשר תפקוד יעיל ויציב של עץ ה-AVL.
* ***parent***– מצביע לאיבר האב בעץ.
* ***left*** – מצביע לבן השמאלי בעץ, הבן בהכרח יהיה קטן ממנו לפי ההשוואה שהסברנו למעלה.
* ***right*** – מצביע לבן הימני בעץ, הבן בהכרח יהיה גדול ממנו לפי ההשוואה שהסברנו למעלה.

**ציור מבנה הנתונים שלנו - CourseManager**

**טבלת ערבול דינאמית מסוג Chain Hashing בשם courses:**עבור הטבלה אנו שומרים את כמות הקורסים המקסימלית שהיא יכולה להכיל כרגע -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 |  |  | | |
|  |  | פונקציית הערבול שלנו: | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  | **עץ דרגות על בסיס AVLבשם** **classes:** העץ ממוין לפי זמן צפייה, לפי מזהה הקורס ואז לפי מזהה השיעור |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

המערך הדינאמי של השיעורים השייכים לקורס זה:

**הוכחת עמידה בסיבוכיות הזמן עבור הפעולות:**

**void \* Init()**

**סיבוכיות זמן הפעולה:** מכיוון שמבנה הנתונים שלנו ריק באתחול,

* כפי שראינו אתחול עץ דרגות על בסיס AVL -
* אתחול טבלת הערבול הדינאמית, אתחול טבלה ריקה -

ולכן סיבוכיות הזמן כוללת תהיה .

**סיבוכיות מקום הפעולה:** מכיוון שמבנה הנתונים ריק ואין רקורסיה, סיבוכיות המקום של הפעולה תהיה .

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

**הסבר מימוש הפעולה:** אנו בודקים האם הקורס כבר קיים במערכת, כלומר בטבלת הערבול שלנו, במידה והוא לא קיים אנו יוצרים קורס חדש. עבור הקורס החדש אנו מקצים מערך דינאמי של שיעורים שמאותחל להיות ריק, ולאחר עדכון המזהה של הקורס כך שיתאים למזהה שקיבלנו, אנו מכניסים אותו לטבלת הערבול שלנו.  
לפני הכנסת הקורס לטבלת הערבול, אנו מפעילים עליו את פונקציית הערבול שלנו – פעולת מודולו בגודל המערך, לאחר מכן מכניסים אותו למקום הנכון בטבלת הערבול או משרשרים במידה והתא תפוס, ומוסיפים 1 לכמות הקורסים שקיימים כרגע במערכת.  
לאחר הכנסה לטבלה אנו בודקים האם ממנה מלאה, ובמידה והיא כן אנו מגדילים את טבלת הערבול להיות פי 2 מגודלה הנוכחי ומכניסים מחדש את הקורסים לטבלת הערבול החדשה.  
  
**סיבוכיות זמן הפעולה:**

* בדיקה האם איבר נמצא בטבלת ערבול דינאמית – בממוצע על הקלט
* הקצאת מקום קבוע עבור המערך הדינאמי הריק של השיעורים השייכים לקורס -
* כפי שראינו בתרגול הכנסת איבר לטבלת ערבול דינאמית מסוג Chain Hashing נעשית בסיבוכיות של - בממוצע על הקלט, משוערך

ולכן סיבוכיות הזמן הכוללת של הפעולה תהיה  **בממוצע על הקלט, משוערך.**

**סיבוכיות מקום הפעולה:**

* הקצאת מערך בגודל קבוע עבור השיעורים -
* במקרה הגרוע הגדלת גודל טבלת הערבול של הקורסים להיות פי שתיים מגודלה הנוכחי -

ולכן מכיוון שהקצנו מקום חדש בזיכרון סיבוכיות המקום הכוללת של הפעולה תהיה .

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

**הסבר מימוש הפעולה:** בודקים האם הקורס שקיבלנו באמת נמצא במערכת, כלומר בטבלת הערבול,   
במידה ולא – מחזירים שגיאה, אחרת ממשיכים ושומרים את הקורס לאחר שמצאנו אותו בטבלת הערבול.  
עוברים על כל המערך הדינאמי של השיעורים השייכים לקורס לזה, במידה והזמן של השיעור גדול מ-0 אנו מסירים אותו מעץ הדרגות הכללי של כל השיעורים, אחרת ממשיכים בבדיקה.  
לאחר שהוצאנו את כל השיעורים המתאימים מעץ הדרגות, נותר למחוק את הקורס מטבלת הערבול הדינאמית.  
כאשר מצאנו אותו בטבלת הערבול שלו מוחקים את כל המידע השמור על הקורס כולל השיעורים, שבשלב זה לא היו שמורים באף מקום למעט במערך הדינאמי של הקורס.  
לאחר הוצאת הקורס, במידה וטבלת הערבול שלנו ירדה ל- מתכולתה המקסימלית אנו מקטינים אותה פי 2 ומכניסים מחדש את הקורסים לטבלת הערבול, על מנת לשמור על יעילות טבלת הערבול הדינאמית שלנו.  
  
**סיבוכיות זמן הפעולה:**

* בדיקה האם איבר נמצא בטבלת ערבול דינאמית – בממוצע על הקלט
* בדיקה עבור mהשיעורים של הקורס האם זמן הצפייה שלהם גדול מ-0, ובמידה וכן הוצאה שלהם מעץ דרגות על בסיס AVL שמכיל בתוכו מקסימום M איברים -
* מציאת ומחיקת הקורס מטבלת הערבול הדינאמית, הקטנת המערך במידה ויש צורך, כפי שראינו פעולה זו נעשית בסיבוכיות של בממוצע על הקלט, משוערך

ולכן אנו עומדים בסיבוכיות הזמן הנדרשת -  **בממוצע על הקלט, משוערך.**

**סיבוכיות מקום הפעולה:**

* הוצאת שיעור מעץ AVL של השיעורים, כאשר עומק הרקורסיה יהיה במקרה הגרוע ביותר בגובה העץ,   
  ולכן סיבוכיות המקום תהיה במקרה הגרוע -
* במקרה הגרוע הקטנת גודל טבלת הערבול של הקורסים להיות חצי מגודלה הנוכחי -

ולכן סיבוכיות המקום הכוללת של הפעולה תהיה .

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

**הסבר מימוש הפעולה:** בודקים האם הקורס שקיבלנו באמת נמצא במערכת, כלומר בטבלת הערבול,   
במידה ולא – מחזירים שגיאה, אחרת ממשיכים ושומרים את הקורס לאחר שמצאנו אותו בטבלת הערבול.  
מגדילים את קאונטר השיעורים שלנו באחד, במידה והקאונטר קטן מהתכולה המקסימלית - מכניסים את השיעור החדש בתא המתאים לו. במידה והקאונטר שווה לתכולה המקסימלית, אנו צריכים להגדיל את המערך הדינאמי פי 2 להכניס את האיברים הקיימים למערך החדש, ואז להכניס גם את השיעור החדש לאחר שהקצנו מקום עבורו.  
  
**סיבוכיות זמן הפעולה:**

* מציאת הקורס הנכון בטבלת הערבול הדינאמית – בממוצע על הקלט
* הוספת השיעור למערך הדינאמי, הגדלת המערך במידה ויש צורך, כפי שראינו פעולה זו נעשית בסיבוכיות של - משוערך

ולכן אנו עומדים בסיבוכיות הזמן הנדרשת -  **בממוצע על הקלט, משוערך.**

**סיבוכיות מקום הפעולה:**

* במקרה הגרוע הגדלת גודל המערך הדינאמי של השיעורים השייכים לקורס להיות פי 2 מגודלה הנוכחי, במקרה וכל הקורסים במערכת הם של הקורס הזה ההגדלה תהיה בסיבוכיות -

ולכן סיבוכיות המקום הכוללת של הפעולה תהיה . **StatusType TimeViewed(void \* DS, int courseID, int classID, int \* timeViewed)**

**הסבר מימוש הפעולה:** נבדוק האם הקורס קיים בטבלת הערבול שלנו, במידה והוא קיים ניכנס למערך השיעורים הדינאמי של הקורס ונמצא את זמן הצפייה שלו בתא המתאים ל-id של השיעור.  
  
**סיבוכיות זמן הפעולה:**

* בדיקה ומציאת הקורס המתאים בטבלת הערבול הדינאמית – בממוצע על הקלט
* מציאת השיעור המתאים במערך הדינאמי של הקורס, השיעור נמצא בתא המתאים ל-id של השיעור ולכן אין צורך לחפש אותו במערך, נעשה בסיבוכיות -

ולכן סיבוכיות הזמן הכוללת של הפעולה תהיה **בממוצע על הקלט**.

**סיבוכיות מקום הפעולה:**

אין הקצאות חדשות ואין רקורסיה ולכן סיבוכיות המקום הכוללת של הפעולה תהיה .

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

**הסבר מימוש הפעולה:** בודקים תקינות וכן האם הקורס קיים במערכת, כלומר האם הקורס נמצא בטבלת הערבול שלנו כפי שמצופה, במידה וכן נשמור אותו.  
לאחר מציאת הקורס הנכון בודקים את זמן הצפייה הנוכחי של השיעור על ידי גישה למערך הדינאמי של השיעורים השייכים לקורס, השיעור בעל classIDנמצא במערך במקום ה-classID.   
לאחר שמצאנו את השיעור במערך - אם הזמן הקודם של השיעור הוא 0, משנים את זמן הצפייה שלו כך שיתאים לזמן הצפייה החדש שקיבלנו ומכניסים אותו לעץ הדרגות של השיעורים (מכל הקורסים) בעלי זמן צפייה גדול מ-0.   
במידה וזמן הצפייה הקודם של השיעור היה גדול מ-0, עלינו תחילה להוציא אותו מהעץ, לתקן את העץ במידה וצריך, לעדכן את השיעור לזמן הצפייה החדש ולאחר מכן להוסיף אותו מחדש לעץ הדרגות של השיעורים.  
השיעורים בעץ הדרגות ממוינים קודם לפי זמני צפייה בסדר עולה, במידה ויש שוויון בזמני צפייה ממיינים לפי מזהה הקורס בסדר יורד, ובמידה ויש שוויון בשניהם ממיינים לפי מזהה השיעור בסדר יורד.  
  
  
**סיבוכיות זמן הפעולה:**

* בדיקה ומציאת הקורס המתאים בטבלת הערבול הדינאמית – בממוצע על הקלט
* מציאת השיעור המתאים במערך הדינאמי של הקורס, השיעור נמצא בתא המתאים ל-id של השיעור ולכן אין צורך לחפש אותו במערך, נעשה בסיבוכיות -
* בדיקת זמן הצפייה הנוכחי של השיעור לפני השינוי והוצאתו מעץ השיעורים בעלי זמן צפייה גדול מ-0 במידה והוא כבר נמצא שם, עץ המכיל מקסימום M שיעורים ומכיוון שזהו עץ דרגות על בסיס AVL סיבוכיות ההוצאה תהיה בגובה העץ ולכן תהיה -
* עדכון זמן הצפייה החדש של השיעור והכנסתו לעץ הדרגות של השיעורים, עץ AVL המכיל מקסימום M שיעורים ולכן הכנסה אליו תהיה בסיבוכיות של גובה העץ -

ולכן אנו עומדים בסיבוכיות הזמן הנדרשת -  **בממוצע על הקלט.**

**סיבוכיות מקום הפעולה:**

* מציאה, הכנסה והוצאה של שיעור בעץ AVL של שיעורים, כאשר עומק הרקורסיה יהיה במקרה הגרוע ביותר בגובה העץ, ולכן סיבוכיות המקום תהיה במקרה הגרוע -

כלומר סיבוכיות המקום הכוללת של הפעולה תהיה .

**StatusType GetIthViewedClasses(void\* DS, int\* courseID, int\* classID)**

**הסבר מימוש הפעולה:** נחפש את השיעור ה-i הכי גדול בעץ הדרגות הממוין של השיעורים.   
נתחיל מהשורש ונבצע פעולה זו עבור כל צומת עד שנמצא את השיעור המתאים: עבור כל צומת נבדוק האם הדרגה של הבן הימני שווה ל-i פחות 1, במידה וכן הצומת הנוכחי הוא הצומת במקום ה-i ונחזיר אותו. במידה ו-i הנוכחי קטן מהדרגה של הבן הימני של הצומת ניכנס לתת העץ הימני ונחפש בו את הצומת במקום ה-i שכן הוא נמצא שם, ונתחיל מחדש את הבדיקה. במידה וה-i הנוכחי שאנו מחפשים גדול מכמות הצמתים בתת העץ הימני ועוד 1 עבור הצומת הנוכחי אנו יודעים כי השיעור ה-i נמצא בתת העץ השמאלי, נפעיל מחדש את הבדיקה עבור תת העץ השמאלי כאשר נחפש בו את הצומת במקום ה-i פחות כמות הצמתים בתת העץ הימני ועוד 1 עבור הצומת שהיינו בו. לאחר שמצאנו את הצומת במקום ה-i בעץ הדרגות הממוין שלנו נשמור את ה- idשל הצומת ונחזיר הצלחה.  
  
**סיבוכיות זמן הפעולה:**

* מציאה של שיעור בעץ דרגות על בסיס AVL של שיעורים, כאשר עומק הרקורסיה יהיה במקרה הגרוע ביותר בגובה העץ, ולכן סיבוכיות הזמן הכוללת של הפעולה תהיה במקרה הגרוע - .

**סיבוכיות מקום הפעולה:**

* מציאה, הכנסה והוצאה של שיעור בעץ AVL של שיעורים, כאשר עומק הרקורסיה יהיה במקרה הגרוע ביותר בגובה העץ, ולכן סיבוכיות המקום הכוללת של הפעולה תהיה במקרה הגרוע -

**void Quit(void \*\* DS)**

**סיבוכיות זמן הפעולה:**

* משחררים את כלל הקורסים והשיעורים במערכת, עובר על כל הקורסים הנמצאים בטבלת הערבול הדינאמית ומשחרר את המערך הדינאמי של השיעורים המתאימים לכל קורס, השיעורים הנמצאים במערכים הם אותם שיעורים הנמצאים בעץ הדרגות ולכן אין צורך לשחרר את עץ הדרגות -

לכן סיבוכיות הזמן הכוללת של הפעולה תהיה ובפרט .  
  
**סיבוכיות מקום הפעולה:**

אין הקצאות חדשות ואין רקורסיה ולכן סיבוכיות המקום הכוללת של הפעולה תהיה .

**הוכחת עמידה בסיבוכיות המקום עבור כלל התוכנית:**

**בכל זמן נתון אנו שומרים במערכת טבלה של קורסים, מערכים בגודל כולל של שיעורים, עץ דרגות של שיעורים בגודל של שיעורים, כלומר עבור שלושתם מידע כולל בגודל .   
כמו כן סיבוכיות המקום של כל הפעולות היא מקסימום .  
לכן סיבוכיות המקום עבור המבנה וכל הפעולות היא - .**