

1 מערכת Boom

האובייקט Boom מורכב מהשדות הבאים :

טבלת הקורסים הראשית

טבלת הקורסים הראשית (בשם *main_table*) היא טבלת ערבול כל תא במערך יכול מצביע לרשימת קורסים. נפעל בשיטת *chain hashing* כאשר פונקציית הערבול היא מודולו גודל הטבלה - דהיינו בהינתן איבר עם מזהה i ניגש לתא ה- $i \% maxSize$.

בנוסף, בכל טבלת ערבול נשמור את הערכים הבאים :

- *currSize* - מספר הקורסים הנוכחי שיש בטבלת הערבול.
- *maxSize* - גודל הטבלה הנוכחית.
- $List \langle Course \rangle * *$ - מערך של מצביעים ל-*List* מסוג *Course*.

כאשר כל *Course* מכיל את המשתנים הבאים :

- *course_id* - מזהה הקורס
- $Hash Table \langle Lectures \rangle lectures$ - טבלת ערבול כאשר כל תא במערך יכול מצביע לרשימה של הרצאות.
- כאשר כל *Lectures* מכיל את המשתנים הבאים :
- *lecture_id* - מזהה ההרצאה המשויך לקורס הנוכחי.
- *time* - זמן הצפייה של ההרצאה הנוכחית.

עץ הדרגות

עץ הדרגות (בשם *time_tree*) הוא עץ AVL מטיפוס *LectureTreeNode* המכיל את המשתנים הבאים :

- *course_id* - מזהה הקורס
 - *lecture_id* - מזהה ההרצאה המשויך לקורס עם המזהה *course_id*.
 - *time* - זמן הצפייה של ההרצאה הנוכחית.
- בנוסף, נשמור בכל צומת בעץ משתנה בשם *rank* המהווה את מספר האיברים של תת העץ של צומת זו כולל היא עצמה.
- כמו כן, הוספנו לעץ את המשתנה *numOfElements* המכיל את מספר האיברים בעץ כרגע.

סיבוכיות הזמן המשווערכת בטבלת הערבול

נבצע הגדלה וכיווץ של טבלת הערבול כפי שלמדנו בהרצאה ע"י שימוש ב-*chain hashing*. פונקציית הערבול שאיתה נשתמש היא מודולו גודל הטבלה. בתחילה, גודל כל טבלת ערבול הוא 7. במידה ואנו רוצים לבצע הגדלה, נגדיל את הטבלה למספר הראשוני הבא הגדול פי 2 לפחות. (למשל, אם גודל הטבלה הוא 17, נקטין את גודל הטבלה להיות 7 כי $8.5 = \frac{17}{2}$ והמספר הראשוני הגדול ביותר הקטן ממנו הוא 7).

במידה ואנו רוצים לבצע כיווץ, נקטין את הטבלה למספר הראשוני הקודם הקטן פי 2 לפחות.

נשים לב כי פעולות ההכנסה, הוצאה וחיפוש של טבלת הערבול מתבצעות ב- $O(\alpha)$ במוצע על הקלט כאשר $\alpha = \frac{n}{m}$ (מספר האיברים בטבלה, m - גודל הטבלה).

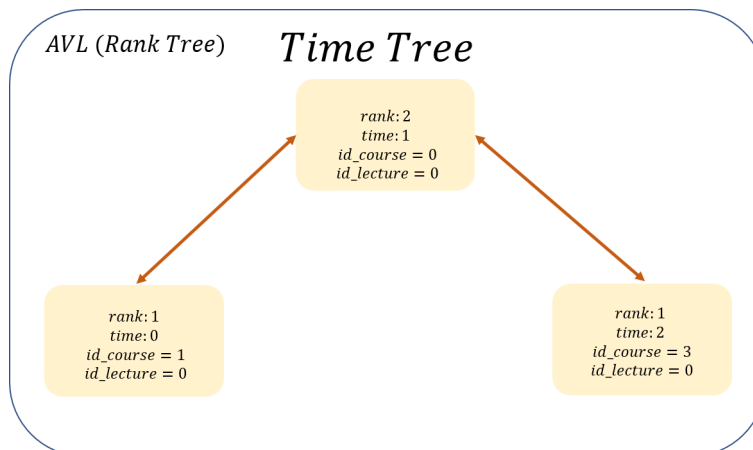
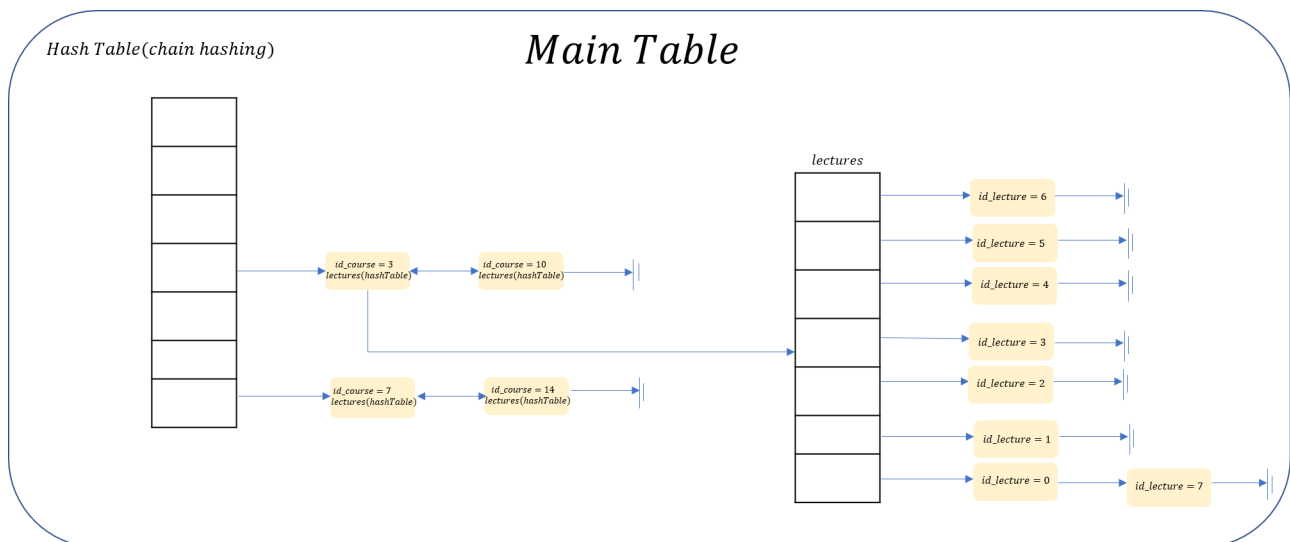
נשים לב כי מתקיים:

$$\alpha = \frac{n}{m} \stackrel{(*)}{=} \frac{O(m)}{m} = O(1)$$

(*) - נשים לב כי $n = O(m)$ כי כאשר מתקיים $2 \cdot \maxSize \leq currSize$ אנו מגדילים את טבלת הערבול לפחות פי 2 וכאשר $\frac{\maxSize}{2} \geq currSize$ מקטינים את טבלת הערבול לפחות פי 2. לכן, יש תלות בין גודל הטבלה למספר האיברים בטבלה ולכן אנו מקבלים כי $n = O(m)$. בכך נקבל כי $\alpha = O(1)$ ולכן פעולות ההכנסה, הוצאה וחיפוש של טבלת הערבול יתבצעו ב- $O(1)$ משוערך במוצע על הקלט.

הערה: נניח שגודל המערך הוא p . כאשר נגדיל את המערך נחפש ראשוני המקיים לפחות שהוא גדול מ- $2p$. לפי השערת ברטראן הראשוני הבא הוא לכל היותר $5p$ (לפי ההשערה עבור כל מספר טבעי n קיים מספר ראשוני ב- $(n, 2n-2)$). לכן, בחיפוש האיבר הראשוני הבא נבצע לכל היותר $3p$ איטרציות, כלומר $O(n)$ כאשר n הוא גודל המערך, ומכיוון ש- $n = O(m)$ אז נקבל כי חיפוש האיבר הבא לוקח $O(m)$, ולכן $O(1)$ משוערך.

להלן איור המתאר את מבנה הנתונים שהוסבר לעיל:



2 מימוש הפונקציות

2.1 פונקציית *init*

- ראשית, ניצור מבנה *DS* - המכיל טבלת קורסים ריקה בגודל 7 ועץ דרגות ריק. נאתחל את *maxSize* להיות 7 ואת *currSize* להיות 0. כמו כן, נאתחל את *numOfElements* להיות 0.
 - במידה של כישלון, נחזיר *NULL*, אחרת נחזיר מצביע למבנה נתונים ריק.
- סיבוכיות זמן: יצירת טבלת קורסים ועץ דרגות ב- $O(1)$.

2.2 פונקציית *AddCourse*

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. במידה והקורס קיים בטבלה נחזיר *FAILURE*.
 - ניצור קורס חדש ונכניס אותו לטבלת הקורסים הראשית ע"י שימוש בפונקציית הערבורל כפי שהגדרנו לעיל.
 - נגדיל את *currSize* ב-1.
 - כעת, נבדוק האם יש צורך להגדיל את הטבלה:
- במידה ו- $maxSize \leq currSize$ נגדיל את המערך למערך בגודל המספר הראשוני הבא הגדול פי 2 מגודל המערך הנוכחי.
- לאחר מכן, נכניס את הקורסים מהטבלה הישנה אל הטבלה החדשה, תוך שימוש בפונקציית הערבורל מודולו גודל הטבלה החדשה, ולבסוף נמחוק את הטבלה הישנה.
- נחזיר ערך הצלחה.

סיבוכיות זמן משוערכת:

1. בדיקת שגיאות - $O(1)$
 2. בדיקה האם הקורס הרצוי קיים - $O(1)$ משוערך.
 3. יצירת אובייקט מסוג *Course* - $O(1)$.
 4. הכנסת האובייקט מסוג *Course* לטבלת הקורסים הראשית - $O(1)$ משוערך.
 5. הקטנה והגדלה של המערך במידת הצורך $O(1)$ משוערך כפי שראינו בתרגול.
- סה"כ סיבוכיות הזמן היא $O(1)$ משוערך.

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה.
- נחפש את הקורס המתאים (מציאת התא המתאים בטבלת הקורסים הראשית ובדיקה ברשימה של התא הנוכחי ע"י שימוש בפונקציית הערובול שהגדרנו לעיל) - במידה ולא קיים הקורס נחזיר FAILURE.
- לאחר הגעה לקורס המתאים ניגש לטבלת ההרצאות שלו ועבור כל הרצאה נבצע את הפעולות הבאות:
 - ניגש לעץ הדרגות $time_tree$ ונסיר את ההרצאה המתאימה.
 - נסיר את ההרצאה מטבלת ההרצאות תחת הקורס הנוכחי.
- מסירים את טבלת ההרצאות תחת הקורס הנוכחי.
- מסירים את הקורס (מטיפוס Course) מטבלת הקורסים הראשי ונקטין את $currSize$ של טבלת הקורסים הראשית ב-1.
- נבדוק האם יש צורך להקטין את טבלת הקורסים הראשית.
 - במידה ו- $currSize \geq \frac{maxSize}{2}$ נקצה טבלה חדשה מגודל המספר הראשוני הגדול ביותר הקטן מ- $maxSize$ לפחות בפי 2. למשל אם גודל הטבלה הוא 17, נקטין את גודל הטבלה להיות 7 כי $8.5 = \frac{17}{2}$ והמספר הראשוני הגדול ביותר הקטן ממנו הוא 7.
 - לאחר מכן, נכניס את הקורסים מהטבלה הישנה אל הטבלה החדשה, תוך שימוש בפונקציית הערובול מודולו גודל הטבלה החדשה.
- נחזיר ערך הצלחה.

סיבוכיות זמן:

- בדיקת פרמטרים - $O(1)$.
 - בדיקה האם הקורס הרצוי קיים בעץ הקורסים הראשי - $O(1)$ משוערך.
 - ע"י גישה לטבלת ההרצאות תחת הקורס הנוכחי, נבצע עבור כל m ההרצאות את הפעולות הבאות:
 - ניגשים להרצאה ה- i ע"י שימוש בפונקציית הערובול (דהיינו לתא ה- $i \% maxSize$) - $O(1)$ משוערך.
 - מסירים את ההרצאה הנוכחית מעץ הדרגות - $O(\log M)$.
 - מעבר על הרשימה הרלוונטית עד הגעה להרצאה הרלוונטית והסרתה - $O(1)$ משוערך.
 - הסרת טבלת ההרצאות המתאימה לקורס הנוכחי - $O(1)$.
 - הסרת הקורס מטבלת הקורסים הראשית - $O(1)$ משוערך.
- סה"כ סיבוכיות הזמן היא $O(m \log M)$ בממוצע על הקלט, משוערך.

2.4 פונקציית AddClass

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים.
 - נחפש את הקורס המתאים (מציאת התא המתאים בטבלת הקורסים הראשית ובדיקה ברשימה של התא הנוכחי ע"י שימוש בפונקציית הערבול שהגדרנו לעיל) - במידה ולא קיים הקורס נחזיר *FAILURE*.
 - אחרת, ניצור הרצאה חדשה ונכניס אותו לטבלת ההרצאות תחת הקורס הרלוונטי ע"י שימוש בפונקציית הערבול כפי שהגדרנו לעיל.
 - נגדיל את *currSize* ב-1 בטבלת ההרצאות תחת הקורס הרלוונטי.
 - כעת, נבדוק האם יש צורך להגדיל את הטבלה:
- במידה ו- $currSize \leq maxSize \cdot 2$ נגדיל את המערך למערך בגודל המספר הראשוני הבא הגדול פי 2 מגודל המערך הנוכחי.
- לאחר מכן, נכניס את ההרצאות מהטבלה הישנה אל הטבלה החדשה, תוך שימוש בפונקציית הערבול מודולו גודל הטבלה החדשה, ולבסוף נמחוק את הטבלה הישנה.

- נחזיר ערך הצלחה.

סיבוכיות זמן משוערכת:

1. בדיקת שגיאות - $O(1)$
 2. בדיקה האם הקורס הרצוי קיים - $O(1)$ משוערך.
 3. יצירת אובייקט מסוג *Lecture* - $O(1)$.
 4. הכנסת האובייקט מסוג *Lecture* לטבלת ההרצאות תחת הקורס הרלוונטי - $O(1)$ משוערך.
 5. הקטנה והגדלה של המערך במידת הצורך $O(1)$ משוערך כפי שראינו בתרגול.
- סה"כ סיבוכיות הזמן היא $O(1)$ משוערך.

2.5 פונקציית WatchClass

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה.
- נחפש את הקורס המתאים (מציאת התא המתאים בטבלת הקורסים הראשית ובדיקה ברשימה של התא הנוכחי ע"י שימוש בפונקציית הערבול שהגדרנו לעיל) - במידה ולא קיים הקורס נחזיר *FAILURE*.
- נחפש את ההרצאה המתאימה בטבלת ההרצאות תחת הקורס המתאים. במידה ולא קיימת הרצאה נחזיר *INVALID_INPUT*.
- במידה ולהרצאה קיים זמן צפייה שונה מאפס (דהיינו, הרצאה זו קיימת בעץ הדרגות), נסיר מעץ הדרגות את ההרצאה עם הזמן שהיה עד כה ונוסיף לעץ הדרגות את ההרצאה עם הזמן החדש הכולל את מזהה הקורס ומזהה ההרצאה בנוסף.
- אחרת, (כלומר להרצאה הנוכחית יש זמן צפייה אפס) נוסיף את ההרצאה עם הזמן החדש הכולל את מזהה הקורס ומזהה ההרצאה בנוסף.
- בנוסף לכך, נגיש לשדה *time* של ההרצאה ונוסיף לו את ערך זמן הצפייה שהתקבל כפרמטר לפונקציה.
- נחזיר ערך הצלחה.

סיבוכיות זמן:

1. בדיקת פרמטרים - $O(1)$.
 2. בדיקה האם הקורס הרצוי קיים - $O(1)$ משוערך.
 3. גישה להרצאה ברשימת ההרצאות תחת הקורס המתאים - $O(1)$ משוערך.
 4. במידה וההרצאה עם זמן צפייה גדול מאפס, נוציא את ההרצאה מעץ הדרגות ונכניס הרצאה עם הזמן המעודכן - $O(\log M)$. במידה וזמן הצפייה הוא אפס הוספת הרצאה לעץ הדרגות לוקחת $O(\log M)$.
 5. שינוי זמן הצפייה של ההרצאה בטבלת ההרצאות תחת הקורס הנוכחי - $O(1)$ משוערך.
- סה"כ כי סיבוכיות הזמן היא $O(\log M)$ בממוצע על הקלט.
- הערה: $O(\log M)$ בממוצע על הקלט, משוערך.

2.6 פונקציית TimeViewed

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה.
- נחפש את הקורס המתאים (מציאת התא המתאים בטבלת הקורסים הראשית ובדיקה ברשימה של התא הנוכחי ע"י שימוש בפונקציית הערבול שהגדרנו לעיל) - במידה ולא קיים הקורס נחזיר *FAILURE*.
- אחרת, נחפש את ההרצאה המתאימה בטבלת ההרצאות תחת הקורס המתאים ונשים ב-*timeViewed* את זמן הצפייה המתאים להרצאה (ע"י גישה לשדה *time* של ההרצאה המתאימה).
- נחזיר ערך הצלחה.

סיבוכיות זמן:

1. בדיקת פרמטרים - $O(1)$.
 2. בדיקה האם הקורס הרצוי קיים - $O(1)$ משוערך.
 3. גישה להרצאה ברשימת ההרצאות תחת הקורס המתאים - $O(1)$ משוערך.
- סה"כ סיבוכיות הזמן היא $O(1)$ משוערך, כנדרש.

2.7 פונקציית *GetIthWatchedClass*

- ראשית, נבדוק שהפרמטרים שהתקבלו בפונקציה אכן חוקיים. במידה ואינם חוקיים, נחזיר שגיאה מתאימה למקרה. כמו כן, במידה ו- $i > numOfElements$ אז נחזיר ערך שגיאה *FAILURE*.
- נחפש בעץ הדרגות את האיבר המתאים ע"י הפעלת פקודת $Select(numOfElements - i + 1)$ (כפי שלמדנו בהרצאה - כאשר $numOfElements$ הוא מספר ההרצאות שזמן הצפייה שלהם שונה מאפס שקיים ברגע נתון). פעולה זו תמצא לנו את ההרצאה במיקום ה- i לפי דרישות הפונקציה.

סיבוכיות זמן:

1. בדיקת פרמטרים - $O(1)$.
 2. הפעלת אלגוריתם $Select$ - $O(\log M)$.
- סה"כ סיבוכיות הזמן היא $O(\log M)$, כנדרש.

2.8 פונקציית *Quit*

- ראשית, נמחק את טבלת הקורסים - נעבור על כל הקורסים ועבור כל קורס נסיר את כל ההרצאות ע"י מעבר בטבלת ההרצאות תחת הקורס הנוכחי והריסת רשימות ההרצאות ומחיקת הטבלה.
- לאחר מכן, נמחק את עץ הדרגות ע"י סיור *PostOrder* ושחרור כל איברי העץ.
- נבצע $DS = nullptr$ ונסיים את התוכנית.

סיבוכיות זמן:

- מחיקת טבלת הקורסים - $O(n + m)$.
 - מחיקת עץ הדרגות - $O(m)$.
- סה"כ סיבוכיות הזמן היא $O(n + m)$, כנדרש.

3 סיבוכיות מקום

במבנה שבנינו קיימים:

- טבלת הרצאות ראשית - n איברים בטבלה המהווים את הקורסים השונים.
- עץ דרגות המכיל לכל היותר m הרצאות (המקרה הגרוע הוא שלכל ההרצאות יש זמן צפייה השונה מאפס).
- הערה: הרצאות עם אפס דקות צפייה לא מאוחסנים בעץ הדרגות.

סה"כ סיבוכיות המקום היא $O(n + m)$