

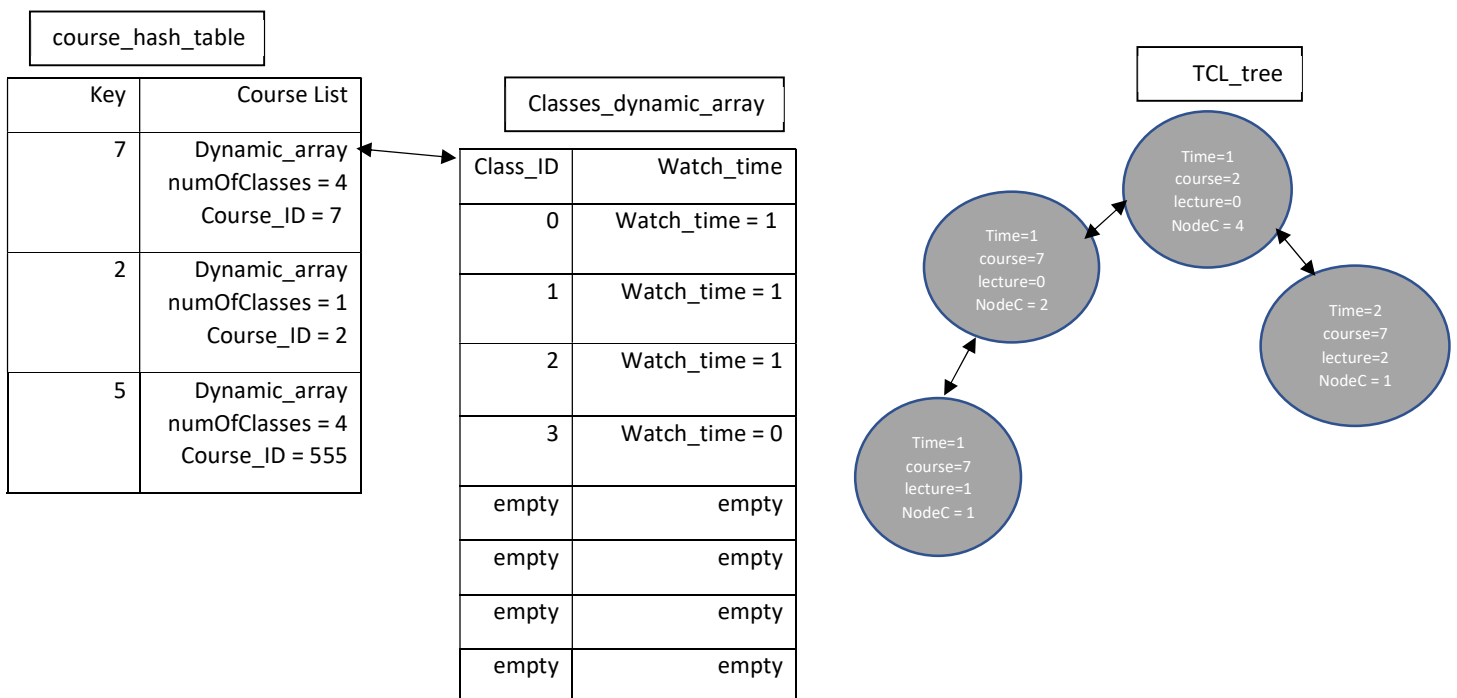
חלק יבש

מבנה נתונים מכיל:

1. `course_hash_table` – hash-table שמייצג את הקורסים שהמפתחות בו הם `course_ID`. הערך בכל תא בhash-table יהיה אובייקט מסוג `course` שבו נשמרים:
 - 1.1. `classes_dynamic_array` – מערך דינמי שיהיה לפחות בגודל מספר ההרצאות בקורס ויגדל במקרה הצורך, ויכיל בכל תא את זמן הצפייה של הקורס (יעבוד בדומה לhash-table)
 - 1.2. `courseID` – מספר הקורס
 - 1.3. `numOfClasses` – מספר ההרצאות בקורס

*hash-table יהיו טבלאות דינמיות עם ערבול רגיל שמשמש בפונקציית מודולו, תוך שימוש בChain-Hashing

2. `tcl_tree(time,course,lecture)` – Rank-Tree - עץ חיפוש בינארי שיכיל את כל ההרצאות עם יותר מ0 זמן צפייה שהמפתחות בו הם `time,course,lecture`, כך שהחשיבות למפתח היא לפי הסדר משמאל לימין(כלומר, קודם `time` אחכ `course` ואז `lecture`).
 - בכל צומת `tcl` נשמר אובייקט עם 3 ערכים מטיפוס `int` – שהם `(time,course,lecture)`.
 - בנוסף כל צומת יכיל `NodeCounter` שיספור את כמות הצמתים בתת העץ שלו.



מימושים לפונקציות:

Void *init()

אתחל tcl_tree, course_hash_table עם פונקציית הערבול המתאימה, כולם ריקים בהתחלה לכן סיבוכיות זמן היא $O(1)$.

StatusType AddCourse (void *DS, int courseID)

נוסיף קורס חדש ל course_hash_table עם 0 הרצאות ונאתחל לו dynamic_array ריק בגודל התחלתי קבוע.

סיבוכיות זמן:

-הוספת איבר ל hash_table $O(1)$, ממוצע על הקלט, משוערך.

-איתחול מערך דינמי בגודל קבוע $O(1)$.

סה"כ: $O(1)$, ממוצע על הקלט, משוערך.

StatusType RemoveCourse(void *DS, int courseID)

נמצא את הקורס המתאים ב course_hash_table נעבור על המערך הדינמי שבתוכו וכך נעבור על כל ההרצאות שלו, אם יש הרצאה עם יותר מ0 זמן צפייה נבצע חיפוש עלייה בTCL_Tree ונמחק אותה מהעץ. לבסוף נמחק את המערך הדינמי ולאחר מכן את הקורס עצמו מה hash_table.

סיבוכיות זמן:

- מציאת הקורס המתאים $O(1)$ ממוצע על הקלט

- מעבר על המערך של ההרצאות ומחיקת כל הרצאה עם יותר מ0 זמן צפייה מהעץ $O(m\log(M))$ במקרה הגרוע

סה"כ: $O(m\log(M))$ בממוצע על הקלט.

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

נמצא את הקורס המתאים ב course_hash_table נוסף עוד הרצאה עם 0 זמן צפייה במערך הדינמי (נגדיל את המערך אם צריך) ונגדיל את המשתנה שסופר את כמות ההרצאות בקורס ב1, לבסוף נכניס ל int* classID את הערך של מספר ההרצאה החדשה.

סיבוכיות זמן:

- מציאת הקורס המתאים $O(1)$ ממוצע על הקלט

- הוספת הרצאה למערך הדינמי $O(1)$ משוערך

- שינוי int* classID והגדלת numOfClasses ב1 $O(1)$

סה"כ $O(1)$ ממוצע על הקלט, משוערך

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

נמצא את הקורס המתאים בcourse_hash_table, לאחר מכן נמצא את ההרצאה המתאימה במערך הדינמי שבקורס, ונוסיף לזמן הצפייה שלה את זמן הצפייה החדש, בנוסף אם זמן הצפייה היה יותר מ0 אז קיים צומת בעץ ההרצאות שמכיל את הרצאה, נמצא אותו ונמחק אותו ונוסיף מחדש עם הערך החדש, אם זמן הצפייה היה 0 אז רק נוסיף צומת חדשה לעץ עם הערך החדש.

סיבוכיות זמן:

- מציאת הקורס המתאים $O(1)$ ממוצע על הקלט
- מציאת ההרצאה המתאימה במערך הדינמי ועדכון ערך זמן צפייה $O(1)$
- מחיקת צומת מהעץ (אם צריך) והוספת צומת חדשה $O(\log M)$
- סה"כ $O(\log(M+2))$ ממוצע על הקלט

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

נמצא את הקורס המתאים בcourse_hash_table, לאחר מכן נמצא את ההרצאה המתאימה במערך הדינמי שבקורס ונכניס את זמן הצפייה שלה למצביע timeViewed.

סיבוכיות זמן:

- מציאת הקורס המתאים $O(1)$ ממוצע על הקלט
- גישה למספר צפיות ועדכון המצביע $O(1)$.
- סה"כ $O(1)$ ממוצע על הקלט.

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

נבדוק קודם אם קיימים לפחות i הרצאות עם יותר 0 צפיות ע"י בדיקת כמות הצמתים שיש בתת העץ של השורש (יש לנו משתנה ששומר את הכמות לכל צומת) אם אין נחזיר Failure אחרת נמשיך: נעבור על העץ של ההרצאות עם יותר מ0 צפיות נתחיל מהשורש ובעזרת האלגוריתם Select שלמדנו נמצא את האיבר הדרוש.

סיבוכיות זמן:

- בדיקה אם קיים לפחות i הרצאות $O(1)$
- מציאת האיבר הדרוש בעזרת אלגוריתם Select $O(\log M)$
- סה"כ $O(\log(M+2))$

void Quit(void **DS)

נעבור על כל תא בHash-Table ונמחק את המערך שבתוכו ואז נמחק את כל טבלת הערבול בנוסף נעבור על כל צומת בעץ ונמחק אותה.

סיבוכיות זמן:

- מחיקת טבלת ערבול $O(n+m)$
- מחיקת עץ $O(m)$
- סה"כ $O(m+n)$

סיבוכיות מקום:

בטבלת הערבול יהיה $O(n)$ תאים ובכל תא יהיה מערך דינמי כך שסכום כל המערכים הדינמיים יהיה $O(m)$ בנוסף בעץ יהיה לכל היותר m צמתים
נקבל סה"כ $O(n+m)$