



מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מגישים : אור אלחדס ודוד ולנסי

החלק היבש – תיאור המערכת האדירה של אור ודוד

המערכת בנויה מהמבנים הבאים:

- עץ splay של כל הגלדיאטורים
- Minimum Heap לשמירת הקבוצות הפעילות בלבד.
- Hash Table לשמירת כלל הקבוצות במערכת.

נפרט תחילה על המבנה עצמו ועל סיבוכיות כל אחד מהרכיבים ולאחר מכן נפרט על כל פעולה בנפרד.

1. **Glad Tree** עץ splay שומר את כל הגלדיאטורים במערכת. הכנסה וחיפוש בעץ splay לוקחת $O(\log n)$ במקרה הגרוע.

2. **group struct** - שומר את מספר הקבוצה ואת האינדקס שבה נמצאת הקבוצה בטבלת ה-
.heap

3. עץ דרגות מיוחד - הוספנו 2 שדות לכל צומת של עץ הגלדיאטורים: מספר הצמתים בתת העץ, וסכום הרמות (score) של האיברים בתת העץ. כדי לשמור על נכונות המידע, שינינו את הפעולות insert/ zig's . במסלול ההכנסה נוסיף 1 ל- num_sons ו- score לשדה sum_scores. כלומר מוסיפים $O(1)$ פעולות על כל צומת בדרך ולא פגענו בסיבוכיות.

בנוסף, בגלגולים אנו מעבירים מידע ע"י גישה לבנים בלבד ולכן אין שינוי בסיבוכיות של גלגול ופעולות אחרות.



מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מגישים : אור אלחדס ודוד ולנסי

4. Hash Table טבלת ערבול מטיפוס Chain Hashing מימשנו אותה ע"י מערך דינאמי של פוינטרים לרשימה מקושרת שתכיל מצביעים ל NodeGroup. כאשר כל NodeGroup יכיל בתוכו את מספר המזהה של הקבוצה, מצביע לעץ Splay מיוחד של הגלדיאטורים של הקבוצה ומצביע ל-group שנשמר בערימת מינימום.

בתור פונקציית ערבול בחרנו מספר מזהה % גודל המערך כפי שראינו בתרגול.

- **Init** – הקצאת מערך באורך n , מאתחלים כל קבוצה בטבלה ע"י המצביעים המוחזרים באתחול ערימת מינימום ועץ ריק.
 - **Insert** – חישוב האינדקס ע"י פונקציית ערבול והוספה לראש הרשימה מקושרת בתא הנכון ב $O(1)$. מוסיפים 1 למספר האיברים שטבלה. אם מספר האיברים גדול מגודל המערך כפול פקטור העומס, מבצעים Rehash ומייצרים מערך חדש בגודל n כאשר n מספר הקבוצות הנוכחי. אנו מעתיקים את כל ה NodeGroups לתאים החדשים ומוחקים את המערך הישן.
כל $Heavy_Factor * size_array$ הכנסות "עולה" $O(1)$ וההכנסה אחרונה $O(Heavy_Factor * size_array)$. כמו שראינו בתרגול, זוהי פעולה שעולה $O(1)$ משוערך.
 - **FindGroup** - נשים לב שעל מנת לבדוק אם קבוצה נמצא או לא, יש צורך בחיפוש ברשימה מקושרת בלבד שהיא באורך $Heavy_Factor$ במוצע כלומר $O(1)$.
5. Minimum Heap כאשר הערכים המוכנסים לערימה הם פוינטרים ל-group.
- **Init** למדנו בתירגול שאיתחול של היפ הוא $O(n)$.
 - **Insert** הכנסה של group חדשה למערכת. ניעזר במערך דינאמי. נסתכל על מקטע עד הפעולה הכבדה. יש לנו m פעולות קטנות של הכנסה ל heap בסיבוכיות $O(\log m)$ ופעולה אחת כבדה של $O(n)$ על כן משוערך יש לנו הכנסה למערכת ב- $O(\log m)$ משוערך.
 - **FindMin** החזרת הקבוצה בעל מזהה הנמוך ביותר ב- $O(1)$ כפי שנלמד בהרצאה.

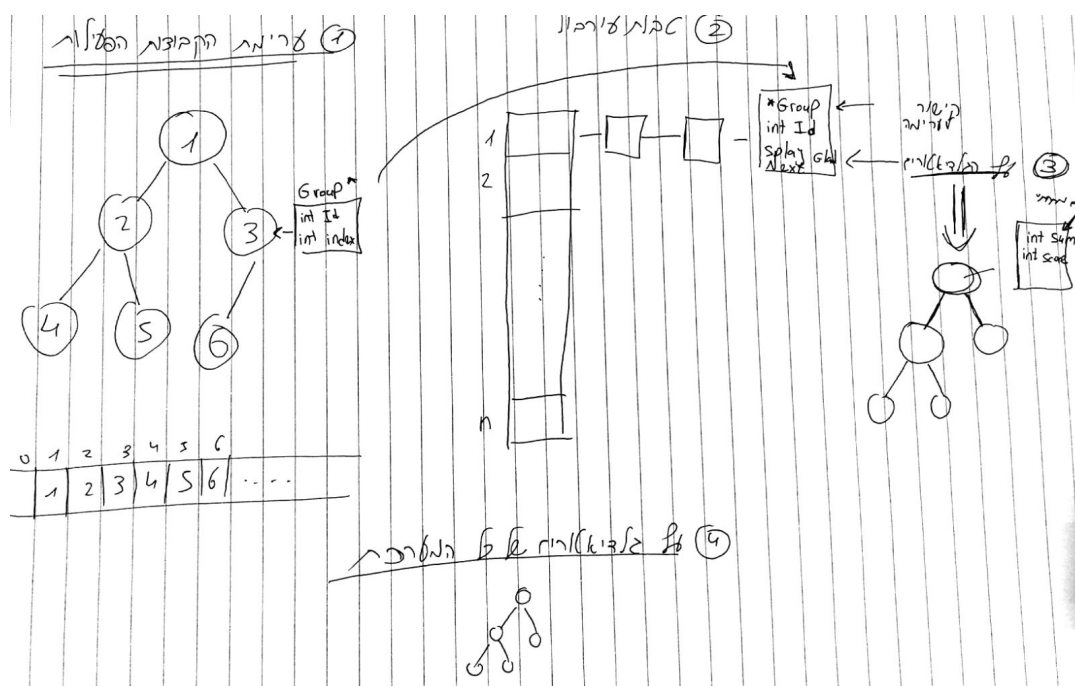


מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מגישים : אור אלחדס ודוד ולנסי

- **DelKey(int)** מאוד דומה לפונקציה שנלמדה בכתה, DelMin, רק שבפונקציה הזו לא מוחקים תמיד את המינימלי אלא מוחקים עפ"י האינדקס שהתקבל בקלט. החסם העליון של פונקציה זו $O(\log n)$

תיאור המבנה שלנו :



נתאר כעת אופן הפעולה של המבנה הנתונים.

• Init –

נעבור על המערך שהתקבל ונכניס למערך שלנו אלמנטים מטיפוס פוינטר group. במקביל, נכניס את הקבוצה לטבלת הערבול יחד עם האלמנט שנוצר ב-heap על מנת שנוכל לקשר בין שני המבנים. בנוסף למערך נשמור שני מספרים, האחד, מספר האלמנטים השמורים בזיכרון בפועל וכמה עוד מקום יש לי בזיכרון עד שהמערך של הערימה מלא. כלל הפעולות הללו לוקחות $O(n)$.



מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מגישים : אור אלחדס ודוד ולנסי


• –addTrainingGroup

תחילה נבצע חיפוש בטבלת הערבול האם הקבוצה נמצאת כבר בסיבוכיות $O(1)$ בממוצע על הקלט. אם לא נמצאה קבוצה כזו נוסיף לערימה קבוצה זו ע"י הפעולה insert שהראנו כי היא בסיבוכיות $O(\log n)$ משוערך כאשר n זה מספר הקבוצות במערכת. במקביל נוסיף את הקבוצה לטבלת הערבול ב- $O(1)$ כפי שהראינו קודם. סיבוכיות זמן: $O(\log n) + O(1)$ worth case + $O(1)$ average משוערך $O(\log n)$ משוערך בממוצע על הקלט.

• –addGladiator

תחילה נפנה לעץ ששומר את כל n הגלדיאטורים ונבצע חיפוש האם קיים גלדיאטור כזה, אם לא נוסיף אותו בסיבוכיות $O(\log m)$. עתה נפנה לטבלת הערבול ונמצא את הקבוצה הרלוונטית ב- $O(1)$ בממוצע על הקלט ולאחר מכן נכניס את הגלדיאטור לעץ דרגות המיוחד ב- $O(\log m^*)$ כאשר m^* זה מספר הגלדיאטורים של אותה קבוצה. חסם העליון של m^* זה $o(m)$ מספר הגלדיאטורים במערכת. סיבוכיות זמן: $O(\log m)$ amortized + $O(1)$ average + $O(\log m)$ amortized בממוצע על הקלט משוערך $O(\log n)$.

• –trainingGroupFight

תחילה נבדוק האם הקבוצות הללו קיימות בחיפוש בטבלת הערבול ב- $O(1)$ בממוצע על הקלט. לאחר מכן נבדוק האם קיימים לשתי הקב  הללו את מספר הגלדיאטורים הדרוש ע"י גישה לעץ הדרגות המיוחד של אותן קבוצות. באמצעות הפעולה findKbest שנעשית ב- $O(\log m)$ כמו שהראנו קודם. נשווה בין הציונים ונבדוק איזה קבוצה להסיר. הסרת הקבוצה תעשה רק על הערימה כלומר נפעיל את פונקציית DelKey ב- $O(\log n)$. סיבוכיות זמן: $O(\log n)$ amortized + $O(1)$ average + $O(\log n)$ worth בממוצע על הקלט משוערך $O(\log n + \log m)$.



מבני נתונים 234218 חורף תשע"ח

גיליון רטוב מספר 2 – מגישים : אור אלחדס ודוד ולנסי

• **getMinTrainingGroup** –

נלך לערימה ונוציא בסיבוכיות של $O(1)$ במקרע הגרוע את האיבר הראשון במערך.

• **quit** –

נעבור על כל המערך של הערימה ונשחרר את הקבוצות הפעולות בסיבוכיות $O(m)$ כמו כן נשחרר את העץ של כלל הגלדיאטורים והעצים של הגלדיאטורים עבור כל קבוצה בשיטת postorder נלך הכי ימינה נמחק ואז שמאלה ואז שורש וכך ב- $O(n)$ נמחק את כל העץ. עתה נמחק את טבלת העירבול, כלומר נמחק ב- $O(n)$ את כל הרשימות המקושרות ואת המערך של הרשימות המקושרות.

סיבוכיות זמן: $O(m)worth + O(m)worth + O(n)worth + O(n)worth$

$$O(m + n)$$

סיבוכיות מקום: בכל רגע נתון נשמור 2 מערכים כדי לשמור מידע על הקבוצות וגודל המערכים הוא לכל היותר קבוע כפול מספר הקבוצות (או אפילו פחות יתכן עבור Hash Table). בנוסף על כך, כל גלדיאטור נשמר פעמיים בלבד. פעם אחת בעץ כללי של כולם ופעם אחת בעץ של הקבוצה המתאימה לו. סה"כ: $O(n + m)$.