

מבני נתונים – תרגיל 3

תאריך פרסום: 30.04.2020

תאריך הגשה: 14.05.2020 23:59

מרצה ומתרגלים אחראים: דקל צור, אריאל אלפרין, נעה בייר

הנחיות:

- הגשת העבודה הינה ביחידים או בזוגות, לבחירתכם.
- העבודה חייבת להיות מוקלדת.
- יש להגיש בפורמט pdf למערכת ההגשה.
- אין צורך להרחיב בדברים שנלמדו במפורש בהרצאה/תרגול, כל טענה אחרת יש להוכיח.
- אם לא נאמר אחרת, כשמדברים על דרישות לזמן ריצה מדברים על ניתוח worst case (זמן הריצה המקסימלי על פני כל הקלטים באותו גודל).
- שאלות לגבי העבודה יש לשאול בפורום באתר הקורס או בשעות קבלה של המרצה/מתרגל האחראיים על העבודה.

שאלה 1 – עצי AVL

סעיף א': הראו שאם x הוא עלה בעץ AVL אז $d(x) \geq \left\lceil \frac{h}{2} \right\rceil$ כאשר $d(x)$ הוא העומק של x ו- h הוא גובה העץ. באילו עצים החסם הדוק?

סעיף ב': הראו שפעולות מחיקה בעץ AVL דורשת $\Omega(\log n)$ סיבובים. כלומר, עליכם להראות משפחה אינסופית של עצי AVL $\{T_n\}_{n \in \mathbb{N}}$ כך שפעולת מחיקה מסוימת בעץ T_n תדרוש לפחות $c \cdot \log(|T_n|)$ סיבובים, כאשר c הוא קבוע חיובי לא תלוי ב- n ו- $|T_n|$ הוא מספר הצמתים בעץ T_n .

סעיף ג': הראו כיצד ניתן, ע"י תוספת זכרון, לתמוך בפעולת $\text{Find}(x, k)$ שתחזיר את k האברים הראשונים שגדולים מהמפתח x בעץ בזמן $O(\log n + k)$. אם מספר האברים עם מפתח גדול מ- x הוא קטן מ- k יש להחזיר את כל האברים האלו.

שאלה 2 - קבוצות

תאר מבנה נתונים השומר זוגות (k, S) כאשר k הוא מספר שלם (המפתח) ו- $S \subseteq \{1, 2, \dots, 10\}$. הקבוצות לא בהכרח זרות. קבוצה S כזו מתוארת על ידי רשימה מקושרת של האברים שלה. הציעו מבנה של זוגות כאלו שיתמוך בפעולות הבאות (n כמות האברים במבנה):

פעולה	תיאור	זמן נדרש
$Init()$	אתחל את המבנה	$O(1)$
$Add(k,S)$	הוסף את (k,S) למבנה.	$O(\log n)$
$Delete(k)$	מחק את הקבוצה עם המזהה (מפתח) k	$O(\log n)$
$First(j,k)$	החזר את המפתח המינימלי מבין הקבוצות שמכילות את j עם מפתח גדול או שווה ל- k . הנח $j \in \{1,2, \dots, 10\}$.	$O(\log n)$

שימו לב שאותה קבוצה יכולה להופיע מספר פעמים עם מפתחות שונים.

שאלה 3 - מיזוג/פיצול

סעיף א': בהינתן שני עצי AVL T_1 בגובה h_1 ו- T_2 בגובה h_2 , כך ש- $\max T_1 < \min T_2$ (כל מפתח ב- T_2 גדול מכל מפתח ב- T_1), ומפתח נוסף x המקיים $\max T_1 < x < \min T_2$, תארו אלגוריתם למיזוג T_1, x, T_2 לעץ AVL חדש בזמן $O(|h_1 - h_2|)$. כלומר, בנו עץ AVL חדש המורכב מכל מפתחות T_1 , כל מפתחות T_2 , ו- x .

הדרכה: אם $h_1 > h_2$, נסו להכניס את x לנקודה המתאימה במסלול הימני ביותר מהשורש לעלה בעץ (המסלול המתקבל מהליכות ימינה בלבד). הראו שהעץ שהתקבל ניתן לתיקון על ידי סיבובים. חשבו מה לעשות במקרה הסימטרי של $h_1 \leq h_2$.

סעיף ב': פעולת $Split(T,k)$ על עץ AVL T ומפתח k מחזירה שני עצי AVL T_1, T_2 כאשר T_1 מכיל את כל המפתחות ב- T שקטנים או שווים ל- k , ו- T_2 מכיל את כל המפתחות ב- T שגדולים מ- k . תארו מימוש לפעולת $Split(T,k)$ בזמן $O(\log n)$ כאשר n כמות האברים בעץ T .

סעיף ג':

תאר מבנה נתונים השומר אברים שלכל אחד מהם יש מפתח וצבע. הצבע הוא שחור או לבן. המבנה צריך לתמוך בפעולות הבאות:

פעולה	תיאור	זמן נדרש
$Add(x)$	הכנס את האבר x (כאשר ל- x יש שדות $x.color$ ו- $x.key$, הניחו שהמפתחות הם מספרים טבעיים)	$O(\log n)$
$Color(k)$	החזר את הצבע של אבר עם מפתח k ($null$ אם לא קיים)	$O(\log n)$
$FlipColors(k)$	החלף את הצבעים של כל האברים במבנה עם מפתח קטן או שווה ל- k . כלומר, אם הצבע של אבר כזה היה לבן הוא יהפוך לשחור ואם הצבע היה שחור הוא יהפוך ללבן.	$O(\log n)$

שאלה 4 – B-trees

סעיף א': בתרגול ראינו מימוש לפעולת מחיקה שמתכנן קדימה, כלומר מבצע MERGE/SHIFT תוך כדי המעבר על העץ, כך שלבסוף כשנגיע למחיקה מעלה נוכל להיות בטוחים שיש בו לפחות t מפתחות. נציע עתה מימוש עצלני למחיקה מעץ B :

- בקריאה ל- $\text{delete}(\text{Key}: k)$ נבצע "מחיקת BST רגילה" של המפתח k . נחפש את k בעץ, על מנת למצוא צומת V המכיל את k . אז אם V עלה פשוט נמחק את k , אחרת נמצא את $k' = \text{Successor}(k)$, נחליף את k ב- k' בתוך V , ונמחק את k' שאנו יודעים שנמצא בתוך עלה (ההכללה הטבעית של מחיקת BST ל-B-tree, שמתחשבת בעובדה שבכל צומת יכול להיות יותר ממפתח אחד).
- אם בעלה שממנו מחקנו יש עכשיו פחות מ- $(t-1)$ מפתחות, בצע תיקונים באופן רקורסיבי (merge/shift מהעלה עד לשורש).

מהו זמן הריצה הטוב ביותר והגרוע ביותר של פעולת delete , במונחים אסימפטוטיים כפונקציה של n ו- t , בשני המימושים. הניחו שמימשנו עץ B כך שכל מפתח מכיל מצביע לצומת שבו נמצא העוקב/קודם שלו (גם במימוש העצלני וגם במימוש שראינו בתרגול).

סעיף ב': תאר את העץ המתקבל אחרי הכנסת $1, 2, \dots, 10$ לעץ B ריק עם פרמטר $t=2$ (הנח שמשתמשים במימוש הלא עצל להכנסה, כלומר ברגע שפוגשים צומת מלא מפצלים אותו). עתה תאר את העץ המתקבל לאחר מחיקת המפתח 2, תחת שני המימושים השונים של אלגוריתם המחיקה (בכל אחד מן המקרים הסבר מדוע זהו העץ שמתקבל, איפה בוצעו פעולות MERGE/SHIFT). למען אחידות הפתרון השתמשו בהנחות הבאות:

- עדיפות לימין: בעת ביצוע, SHIFT נעדיף להשאיל מהאח הימני במידת האפשר. באופן דומה, בביצוע MERGE נעדיף להשתלב עם האח הימני (כלומר, אם שני האחים מועמדים קבילים, נעדיף את הימני).
- עדיפות לעוקב: כשנרצה להחליף ערך בעוקב או קודם, נעדיף להחליף בעוקב.

סעיף ג': נניח שבעץ B עם פרמטר t (דרגה מינימלית) זמן הקריאה מהדיסק הוא $\alpha + \beta t$ (כשמבקשים לקרוא את התוכן של צומת כלשהו, זה הזמן שיקח לנו לקבל אותו). מה זמן החיפוש הכולל ומהו t אשר ימזער אותו (נסתפק במשוואה ל- t שתלויה ב- α, β , אין צורך להגיע לפתרון מדויק). הניחו שמתעניינים רק בזמן קריאה מהדיסק, כלומר תוכלו להזניח זמן מעבד.