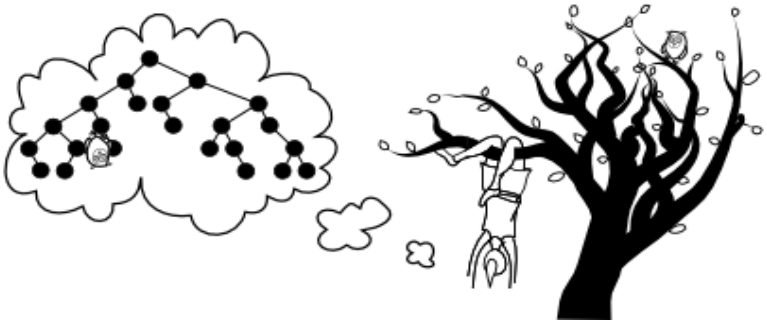
מבני נתונים – תרגיל 4

מגישים: **אסלאן אסלאן**
עידן קלבו



חלק א׳ – תיאורטי

שאלה 1 :

נתון Tree-B עם דרגה מינימלית, t המכיל n צמתים. כל הצמתים, כולל השורש, מלאים .בצמתי העץ מאוחסנים בלוקים של קובץ, כך שגודל כל בלוק הוא .D חשבו את היחס)כביטוי של(D,t,n בין המקום הדרוש לאחסון ה Tree-B והמקום הדרוש לאחסון ה Tree-B-Merkle הנגזר ממנו .הניחו שגודל מצביע לצומת זה בייט אחד, ושבעלים קיימים מצביעים אך הם מצביעים ל .null ותזכרו שהפלט של SHA1 הוא 20 בייטים.

תשובה: עבור B-TREE עם דרגה t , n צמתים מלאים נסיק שהוא מכיל 2n-1 מפתיחות בכל צומת, יש לנו n צמתים כאלה וכל צומת יש בלוק בגודל D בנוסף לזה יש את החצים שכל אחד מהם הוא בגודל של 1 ביט , יש לנו 2t בנים לכל צומת כפול n צמתים לכן בסכה כל נקבל

Size-B-tree=(2t-1)*D*n+2t*n

עבור MBT , כיוון שיש לנו מבנה זהה לזה של BT החישוב דומה רק שכל צומת היא בגודל 20 ביט במקום D(2t-1) לכן נקבל

Size-MBT=20*n+2t*n

לכן היחס הוא : (2t-1)*D*n+2t*n\20*n+2t*n

שאלה 2:

נתונים Tree-B ולצדו ה MBT שנגזר ממנו .המטרה היא לשמור חתימה עדכנית עבור ה Tree-B בכל רגע נתון .האם עדכון ה Tree-B ע״י הכנסת בלוק חדש או מחיקת בלוק קיים, מחייב חישוב מחדש לכל צמתי ה ?MBT אם כן - הוכיחו .אם לא - נסחו מחדש את אלגוריתמי ההכנסה והמחיקה של)Tree-B ותוסיפו שדות לצמתי העצים במידת הצורך ,)כך שיכללו עדכון שיטתי ויעיל ל, MBT המצריך חישוב מחדש רק לחלק מצומצם מצמתי ה .MBT חשבו את סיבוכיות הזמן והמקום של האלגוריתמים שניסחתם מחדש.

תשובה: לא

אנחנו מדברים על עץ חתימות מקביל לזו של BT ז"א כל שינוי בעץ ה BT מחייב שינוי ב עץ החתימות המקביל MBT , כלומר עבור פעולות כגון הוספה הסרה לעץ BT אנחנו נדרשים לעדכן במקביל את עץ החתימות שלנו שנגזר מעץ ה BT , איפה העדכון הזה יתבצע ותחת אילו תנאים?
העדכון יהי בצומת שעבר שינוי ומשם פשוט לעדכן את הבנים של אותה צומת אשר נעשה בה שינוי

- למשל עבור מחיקה יהיה קל למחוק מצומת שיש בה לפחות T-1 מפתחות
- הקושי יהיה במחיקה מצומת שיש בה למשל T-1 מפתיחות

כלומר אנחנו צריכים לסדר את העץ BT ומקביל לו את עץ החתימות כך שישמור על זהותו כעץ BT

לכן בשביל לעשות את זאת אנו נחזיק משתנה חדש נקרא לו int dirtyleaf שיצביע על מיקום העלה מבין האחים שלו זאת אומרת זה משתנה שרץ מ 0 עד אינסוף במקרה של עץ אינסופי(לא פיזקלי אבל נניח) ועבור המספרים האלה נצור לנו מסלול שורש עלה שעבורו נצטרך לעדכן את הצמתים המקבילים ב MBT .
עבור כל פעולה אם זה מחיקה או הוספה נצטרך לאחסן בחתימה של הפונקציות את המשתנה הנ"ל של הצמתים שנכנסו במסלול שורש עלה,
אחרי זה נצטרך לעדכן את פונקציות ה Shifting ופונקציות Merging

פונקציה MERGING תעדכן לנו את המשתנה הנ"ל בכל צומת כך שבסוף הריצה מסלול שורש עלה יהיה מתועד,
פונקציה SHIFTING היא תהיה אחראית על עדכון צומת ה MBT בסוף ריצת התוכנית על ידי העברת בלוקים מצומת לאחרת .

סיבוכיות זמן ומקום

המסלול הארוך ביותר מכיל log(n) צמתים ובכל צומת יש לנו K תווים לכן סיבכיות המקום היא

O(log(n))

עברו סיבוכיות זמן ריצה של הכנסה המקרה הגרוע הוא שנגיע עד לעלה ובחזרה נתחיל לעדכן את העץ MBT כלומר זמן הריצה תלוי בגובה העץ (כיוון שעץ ה MBT תלוי במספר הבלוקים שצריך לעדכן ובגובה העץ המושרש בכל צומת) משמע הדבר שעדכון עץ ה MBT יהיה שווה לסכום גדלי תתי העצים המושרשים בו.
פונקציית Shifting תעדכן לי צומת כאילו במקרה קצה נבצע עדכון בכל צומת בכל רמה אפשרית שזה שכול ל log(n) פעולות בסה"כ זה n+log(n) =~ $O(n)$.

פונקציית Merging מעדכנת בנוסף לצומת גם את הצמתים האחים, לכל היותר $(2t-2)\cdot\log n$ פעולות=$O(n)$.

לכן נקבל את הנוסחה הבאה

$$T(n,t) = \sum_{k=0}^{\log(n-1)} [2 + t^k \cdot (2t-1)] = 2\log(n-1) + (2t-1)\cdot \sum_{k=1}^{\log(n-1)} t^h$$
$$= 2\log n + (2t-1)\cdot \frac{t^0 \cdot \left(t^{\log(n-1)} - 1\right)}{t-1}$$
$$= 2\log n + \frac{(2t-1)\cdot(n-1-1)}{t-1} \in O(n)$$

שאלה 3 :

הסיבה שעבורה הם בחרו להשתמש במשתנים קבועים היא על מנת לשמור על היחיודת של האלגוריתם כלומר הם רצו שיהיה חד ערכו במובן הזה , והפלט שלו יהיה פלט רציף, משמע הדבר שעבור שני קלטים שונים נניח X Y הפלט אמור להיות שונה עבור שני הערכים הספציפיים האלה , בניגוד לזה שאם יבחרו אקראית יתכן מצד שעברו שני קלטים שונים נקבל את אותו הפלט.