חלק א' – תיאורטי

מגישים:

יואב אלמעלם 204256655

רון אלקלעי 308197235

1) BTree- נשים לב כי כל צומת מכילה 2t-1 בלוקים בגודל D ולכל צומת קיימים 2t מצביעים בגודל בייט אחד. ולכן המקום הדרוש לשמור את העץ הוא- n(2t-1)D + n(2t)

MBT- כל צומת מכילה את פלט הפונקציה SHA1 ולכן גודלה הוא 20. בנוסף לזאת לכל צומת קיימים 2t בנים וכל מצביע בגודל בייט אחד ולכן המקום הדרוש ולשמור את העץ הוא- 20n + n(2t)

ולסיכום –

(n(2t-1)D + n(2t)) / (20n + n(2t)) =

((2t-1)D + 2t) / (2t + 20)

2) נציע אלגוריתם המחשב מחדש רק חלק מצומצם מצמתי הMBT מכיוון שאין צורך בחישוב כולל של כל צמתי העץ עבור הכנסה\מחיקה של בלוק מהעץ.

ראשית נוסיף מצביע מכל צומת בBT וMBT אל האבא שישמש אותנו גם עבור ההכנסה וגם עבור המחיקה מהעץ- כמובן שעבור כל שינוי של הוספה\פיצול\מחיקה\איחוד וכדומה נעדכן את המצביעים לאבא ולילדים כנדרש.

<u>הכנסה:</u> נשים לב כי כל צומת בעל 2t-1 מפתחות בדרך גורר לפיצול הצומת. פעולה זו יוצרת צורך בשינוי החתימה של 3 צמתים- האבא ושני הילדים החדשים. נחשב מחדש, עבור כל פיצול וברגע הפיצול, את החתימה עבור הצומת השייכת לתת העץ בו לא מתבצעת ההכנסה (אל האב והבן השני נחזור בהמשך). לאחר ההוספה נחשב את החתימה עבור העלה בו הוכנס הבלוק החדש ואת כל האבות של העלה בדרך לשורש (במסלול זה נעבור גם אצל כל "בן שני" שלא חישבנו את החתימה החדשה עבורו ואל האבא מהפיצול)

<u>מחיקה:</u> נבדיל בין 2 תהליכים שונים- א. הדרך אל הצומת ממנה אנו רוצים למחוק בלוק , ב. מחיקת הבלוק עצמו.

א. בדרך אל הצומת אנו עלולים להתמודד עם שתי פעולות אשר ישנו את החתימה עבור העץ – איחוד בין 2 אחים , והשיפטינג בין 2 האחים.
במקרה של שיפטינג נחשב מחדש באופן מידי את חתימת האח ממנו נלקח הבלוק (אל האח השני והאבא נחזור בהמשך במסלול חזור מהעלה).
במקרה של איחוד בין שני אחים אין צורך לעדכון מידי של חתימת הצומת (נחזור אל הצומת והאבא בהמשך במסלול חזור מהעלה).

ב. בתהליך מחיקת הבלוק עצמו, אנו עלולים להתמודד עם 4 מקרים שונים-
1. הבלוק נמצא בעלה- מחק את הבלוק מהעלה וחשב מחדש את החתימה של כל צומת מהעלה עד השורש.
2. הבלוק נמצא בצומת פנימית ולילד השמאלי של הצומת יש לפחות t מפתחות – נשמור ונמחק את הערך המקסימלי של תת העץ היוצא מהבן השמאלי (מחיקה לפי 1.) (עבור כל פעולה בדרך נשנה את החתימה כפי שהוצג בחלק א.). מהעלה בו נמצא הערך המקסימלי נחשב את החתימה מחדש עבור כל הצמתים עד הצומת המבוקשת , נחליף את הבלוק המקורי שאנו רוצים למחוק עם הערך המקסימלי שכבר מחקנו, ונחשב מחדש את כל החתימות מהצומת הנ"ל עד שורש העץ.

3. הבלוק נמצא בצומת פנימית, לילד השמאלי של הצומת אין לפחות t מפתחות, ולילד הימני של הצומת יש לפחות t מפתחות – נפעל באותה דרך כמו ב2. רק שכעת נחפש את הערך המינימלי בתת העץ הימני.
4. במקרה בו לשני הילדים יש t-1 בלוקים בלבד- נאחד אותם, מה שיגרום להורדת הבלוק המבוקש לצומת , נמחק את הבלוק לפי 3\2\1 והחתימה תשתנה בצומת זה ובכל האבות עד לשורש כפי שהוצג קודם.
5. במקרה והבלוק המבוקש אינו נמצא בעץ כלל, הליך החיפוש הסתיים בעלה ולכן נעדכן את החתימות רק באבותיו של העלה עד השורש.

(אין צורך לשנות את חתימת העלה מפני שבוודאות אינו השתנה).

חשוב לציין בדגש כי עבור הכנסה ומחיקה של בלוק, נחשב תמיד את החתימה מחדש מהעלה עד שורש העץ.

#חישוב זמן הריצה: (נניח זמן הריצה של SHA1 קבוע)

זמן הריצה של פעולות ההכנסה הוא O(logn) כאשר n מספר הצמתים בעץ. במקרה הגרוע ביותר בהכנסה- נבצע O(logn) פיצולים, על כל פיצול נצטרך לחשב פעמיים את פונקציית ההאש- פעם אחת עבור הצומת אשר אינה נמצאת בדרך, ופעם שניה שתחושב בדרך חזרה מהעלה לצומת. נשים לב כי עבור כל שינוי בעץ, עדכון ה MBT ייקח אותו סדר גודל של זמן(כפולה בקבוע) ולכן סך הכל קיבלנו:

O(logn) על פעולת ההכנסה + O(2 logn) עבור ההאשינג = O(logn).

זמן הריצה של פעולות המחיקה הוא O(logn) כאשר n מספר הצמתים בעץ. במקרה הגרוע ביותר, עבור כל צומת שנעבור נצטרך לבצע שיפטינג כך שנצטרך בעצם לבצע O(2logn) עדכונים של חתימת ההאשינג עבור הצמתים שעברו שיפט ועבור המסלול מהעלה עד השורש ולכן סך הכל קיבלנו:

O(logn) עבור המחיקה עצמה + O(2logn) עבור ההאשינג = O(logn).

#הוספת זכרון - הוספנו רק מצביע לאבא של צומת ולכן עבור 2 העצים יחדיו הוספנו רק 2n * bits לזכרון.

הזכרון הדרוש לBT, MBT הוא O(n) .

בהוספת בלוק חדש הזכרון גדל בגודל הבלוק + ביט אחד כפול כמות הפיצולים בדרך = לכל היותר O(logn) זכרון.

במחיקת בלוק מהעץ אין אנו מוסיפים מצביעים או צמתים חדשים לעץ אלא רק מבצעים פעולות "הזזה" או צמצום ולכן דורשמת מקום קבוע של O(1).

3) מתכנני 1SHA בחרו לאתחל את המשתנים לערכים קבועים ולא בערכים רנדומליים המוגרלים כל פעם מחדש מכמה סיבות-

ראשית האורך המקסימלי של ההאשינג נקבע ע"י כמות הביטים של הערכים ההתחליים.

בנוסף לזאת כדי שהפלטים של הפונקציה יהיו עקביים, על הערכים של הפונקציה להיות קבועים.

ולסיום, והכי חשוב- מטרת החשיפה, של אותם 5 משתנים קבועים, נועדה למטרת שחזור הקובץ ע"י בדיקה- כלומר הרצת הפונקציה על קלט מסוים, תחזיר את אותו הפלט בכל פעם. כמו למשל בדוגמא שניתנה בתחילת העבודה- כדי לוודא כי קובץ האקליפס שהורד תקין ואינו זדוני\ עבר שינוי כלשהו, ניתן יהיה לקבל את אותו הפלט עבור בדיקת הקובץ שהורד והפלט שאמור היה להתקבל המוצג באתר אקליפס.