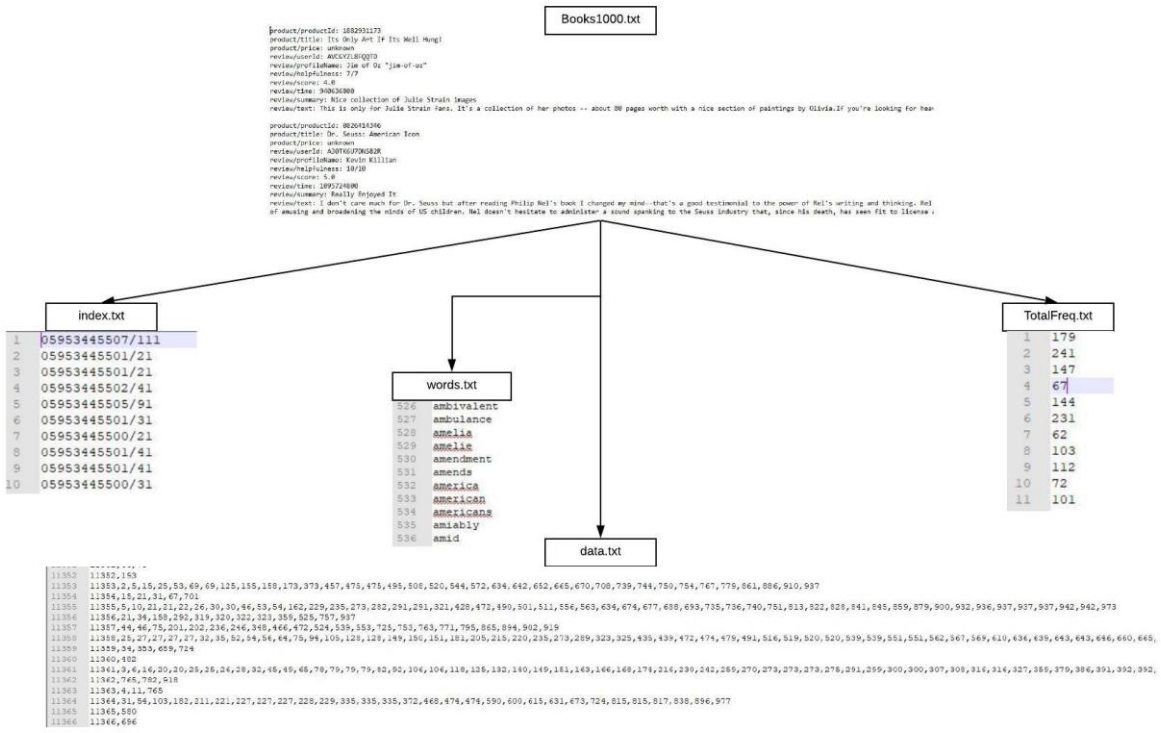
Analyzes

פרטי האינדקס:

כאשר מריצים את התוכנית מתווצר ארבעה קבצים בינאריים בפורמט הבא:

- קובץ index.txt המכיל בכל שורה שלושה נתונים(מספר המזהה של כל review הוא מספר השורה בקובץ):
  - productId: האורך שלו קבוע (10 תווים) והוא נשמר בתחילת הקובץ.
  - Score: בגודל קבוע (תו בודד) הוא מספר של Byte1.
  - Helpfulness: נשמר בצורה של מחרוזת בתוך המחרוזת יש "\" כדי שנוכל להפריד בין המונה והמכנה וברגע שקוראים אחד מהם אנו ממירים אותו ל-integer.
- קובץ TotalFreq.txt שמכיל בכל שורה אורך של כל Review (מספר המזהה של כל review הוא מספר השורה בקובץ).
- קובצ ה-words.txt מכיל את כל המילים שנמצאים בכל ה-reviews ללא חזרה כך שכל מילה נמצאת בשורה בודדת.
- קובץ ה-data.txt מכיל את הנתונים של המילים הממוינים כך שכל שורה קשורה למילה בקובץ ה-words לפי הסדר שלהם ,כל שורה מכילה מספר האינדקס של המילה בקובץ words.txt (מספר השורה) לידיה מספר כל ה reviews מופרדים בפסיקים.



איך מיצרים את האינדקס(IndexWriter):

1. קוראים Buffer בגודל של 1 ג'יגה בייט מהקובץ של ה Books שנמצה ב תיקיה שמקבלת אותה השיטה.
2. מיצרים את הקבצים שלנו שהם index.txt,TotalFreq.txt,Text.txt
3. עוברים על ה buffer ומנתחים אותו ושומרים את הנתונים שצריכים במילון, גודל המילון יהיה בין 500-700 MB וכרגע הזיכרון של המחשב תופס GB 1.7.

4. אחרי ששמרנו את כל הנתונים הרצויים מה-Buffer במילון אנחנו משחררים אותו, אז ברגע זו הזיכרון תופס 0.7 GB במצב הגרוע.
5. עוברים על המילון שלנו וכותבים את המידע בקבצים שייצרנו (קובץ ה- Text.txtאנו כותבים בו את כל הטקסטים)
6. כאשר אנחנו עוברים על המילון מיצרים עוד מילון ושומרים בו את המילים וכל נתון ששומרים אנו משחררים אותו מהזיכרון
7. חוזרים שוב לנקודה 1 עד שמסיימים ניתוח כל הנתונים בקובץ.
8. אחר כך אנחנו ממינים את מילון המילים לפי האלפבית ונותנים לכל מילה ID לפי הסדר ושומרים אותם ב-words.txt
9. ואז לפי אלגוריתם ה-BSBI יצרנו קבצי זוגיות כך שכל קובץ הוא בגודל של 3 ג'יגה, ועל ידי מיון מיזוג אנו מזגנו כל שני קבצים עד שקבלנו את כל הנתונים בקובץ אחד.
10. בסוף עברנו על קובץ שקיבלנו יצרנו את קובץ ה data.txt.

ניתוח זמן הריצה של IndexWriter :

1. לעבור על קובץ הBooks ---> O(n/bufferSize)
2. ניתוח ה- buffer --> O(n)
3. כתיבת הנתונים שנמצאים במילון הראשון --> O(n)
4. מיון המילון של המילים --> O(NLogN) במצב הגרוע
5. יצירת קבצי הזוגיות--> O(n*n) (לעבור על הטקסטים ולמצוא את ID המילה)
6. מיון מיזוג :
    a. למזוג שני קבצים --> O(m+n) m:lines in file1,n:lines in file2
    b. לחזור שוב ושוב עד שנקבל קובץ אחד--> O(n/2)
7. יצירת קובץ ה-data.txt מהזוגיות--> O(n)

תכונות המחשב:

מערכת ההפעלה: Windows 10

זיכרון(RAM): 8 GB

CPU : 3.41 GHz

סוג הדיסק: SSD

קריאת הנתונים מהזיכרון/הדיסק

כאשר מריצים את התוכנית שום דבר לא נשמר בזיכרון, וברגע שצריכים לקרוא נתונים מסוימים ניגשים לקובץ המתאים בדיסק, קוראים buffer של חצי גיגה ובגלל שהנתונים הם מסודרים אין צורך לעבור על ה-buffer כדי לדעת אם הידע הרצוי הוא נמצה בו או לא.

אם ה-buffer מכיל את המידע אנו עוברים עליו בעזרת החיפוש הבינארי,ואם לא קוראים החצי ג'יגה הבאה.

הזמן שלוקח להריץ שאילתה אחת הוא: 0.2-1 min

הזמן שלוקח להריץ 100 שאילתות אקראיות של getReviewsWithToken ו 100 שאילתות אקראיות של TokenFrequency. הוא 90 דקות.

הגודל הצפוי של האינדקס

אפשר לחשב גודל האינדקס דרך הנוסחאות הבאות:

- קובץ ה-index.txt: שומרים productid, score, helpfulness בגדלים קבועים שהם 10, 1, 3-5 בהתאם לכן הנוסחא היא 16 כפול מספר ה-review.
- קובץ TotalFreq.txt בכל שורה שומרים מספר מסוג integer שגודלו 3 byte במומצע אז הנוסחא היא 3 כפול מספר ה-reviews.
- קובץ ה-words: ממוצע האותיות במילה הוא 7 אותיות, גודל השורה היא 2 bytes לכן הנוסחא: (2bytes* מספר ה-reviews) + (7 * מספר המילים ללא חזרה)
- קובצ ה-data.txt: עבור כל מילה שומרים מערך כך שכל מערך מכיל מספרים של ה-reviews שהופיעה בהם המילה, הנוסחא היא:
  מספר המילים * (2 bytes + 3 bytes * מספר ההופעה למילה)