שאילתות באינטרנט

תרגיל 2 – דחיסת רשימות התפוצה

1. דרישות התרגיל

בתרגיל זה עליכם לממש את רשימות התפוצה.

את רשימות התפוצה יש לדחוס בשיטת ה Varint.

זכרו כי יש לקודד את <u>ההפרשים</u> בין מספרי המסמכים ולא את מספרי המסמכים עצמם. בנוסף למספר המסמך יש לשמור גם את מספר הפעמים שהמילה מופיעה במסמך.

רשימות התפוצה יהיו מאוחסנות אחת אחרי השנייה בקובץ אחד בשם text.pl.

עבור כל מילה במילון, עליכם להוסיף את המצביע לרשימת התפוצה כלומר את מיקום התחלת הרשימה בקובץ.

המצביע לרשימת התפוצה יהיה בגודל של 4 בתים ויופיע עבור כל מילה אחרי גודל התחילית המשותפת. לאחר הוספת המצביע לכל מילה, גודל כל שורה בטבלה יגדל ב 40 בתים ויהיה בגודל 102 בתים.

בזמן שאילת שאילתה הדורשת מידע מתוך רשימת התפוצה, עליכם למצוא במילון את המצביע לרשימת התפוצה הרלוונטית ולקרוא רק אותה מהדיסק אל הזיכרון.

2. דרישות הקוד

התכנית תכיל לפחות את שתי המחלקות הבאות: (ככל הנראה התכנית תכלול מחלקות נוספות הנחוצות לצורך מימושן)

2.1. SecondIndexWriter: בהינתן נתונים גולמיים, המחלקה תיצור את הקבצים הנדרשים על הדיסק בספריה שתתקבל כקלט.

בתרגיל זה ניתן להניח כי כאשר בונים את הקבצים, כל הנתונים יכולים להיות מאוחסנים בזיכרון.

המחלקה מאפשרת גם למחוק את הקבצים מהדיסק על ידי מחיקת כל הקבצים מהספרייה.

2.2. SecondIndexReader: לאחר שנוצרו הקבצים על הדיסק ניתן להשתמש במחלקה כדי לגשת למגוון של נתונים. כלומר מבנה הקבצים צריך לתמוך במימוש יעיל של המתודות המוגדרות. ניתן להניח כי המתודות יופעלו רק לאחר שהאינדקס ייבנה על ידי SecondIndexWriter.

תיאור של הממשק שצריך להיות ממומש מתואר בעמודים הבאים.

```
class SecondIndexWriter:
     def __init__(self, inputFile, dir):
          """Given product review data, creates an on
          disk index
          inputFile is the path to the file containing
          the review data
          dir is the path of the directory in which all
          index files will be created
          if the directory does not exist, it should be
          created"""

     def removeIndex(self, dir):
          """Delete all index files by removing the given
          directory
          dir is the path of the directory to be
          deleted"""
```

```
class SecondIndexReader
     def __init__(self, dir):
          """Creates a FirstIndexReader object which will
          read from the given directory
          dir is the path of the directory that contains
          the index files"""

     def getTokenFrequency(self, token):
          """Return the number of reviews containing a
          given token (i.e., word)
          Returns 0 if there are no reviews containing
          this token"""

     def getTokenCollectionFrequency(self, token):
          """Return the number of times that a given
          token (i.e., word) appears in all the reviews
          indexed (with repetitions)
          Returns 0 if there are no reviews containing
          this token"""

     def getReviewsWithToken(self, token):
          """Returns a series of integers of the form id-
          1, freq-1, id-2, freq-2, ... such
          that id-n is the n-th review containing the
          given token and freq-n is the
          number of times that the token appears in
          review id-n
          Note that the integers should be sorted by id
          Returns an empty Tuple if there are no reviews
          containing this token"""

     def getNumberOfReviews(self):
          """Return the number of product reviews
          available in the system"""

     def getTokenSizeOfReviews(self):
          """Return the number of tokens in the system
          (Tokens should be counted as many times as they
          appear)"""
```

3. בדיקת התרגיל

בבדיקת התרגיל ייעשה שימוש בערכת נתונים קטנה בסדרי גודל של הערכות הנמצאות באתר הקורס.

התרגיל ייבדק שאכן המימוש נותן מענה נכון לשאילתות וכן שפורמט הקבצים תואם את הדרישות.

בדיקת התרגיל נעשית בעזרת מערכת אוטומטית. כדי שבדיקת התרגיל שלכם לא תיכשל (ותגרום להורדה בציון) הקפידו היטב על שמות המחלקות ועל מימוש הממשק.

4. הגשת התרגיל

- התרגיל יוגש דרך אתר המכללה בפורמט ZIP.
- עבור כל זוג יש להגיש רק הגשה אחת. שם הקובץ צריך להיות ID1_ID2.zip כאשר ID1 ו ID2 הם מספרי הזהות של הסטודנטים המגישים את הפרויקט.
- **קבצי הקוד צריכים לכלול שני קבצים עם השמות SecondIndexWriter.py ו SecondIndexReader.py. תכנית הבדיקה תייבא (import) קבצים אלו כך שחשוב שתקפידו על השמות הנכונים (כולל אותיות גדולות וקטנות) קבצים אלו יכילו את המחלקות שנדרשתם לפתח.**
- **במידה והפרויקט שלכם מכיל קבצי קוד נוספים (מחלקות נוספות), באחריותכם לייבא אותם (import) מתוך הקבצים FirstIndexWriter.py ו FirstIndexReader.py.**
- יש לוודא כי הקובץ שהועלה הוא בפורמט הנכון ומכיל את כל הקבצים הרלוונטיים.

דוגמה לקידוד רשימות התפוצה

ab	3,8	700,1	
abc	3,3	5,2	
ba	999,5	1000,500	70,000,7
bcabc	. . .		
bcacc	. . .		
bdd	. . .		

רשימת המספרים לקידוד אחרי תושלם הפרשים:

$\underbrace{3, 8, 697, 1}_{ab}, \underbrace{3, 3, 2, 2}_{abc}, \underbrace{999, 5, 1,500, 69000, 7}_{ba}$

תוכן קובץ רשימת התפוצה

$\underbrace{10000011}_{3}, \underbrace{10001000}_{8}, \underbrace{00000101\ 10111001}_{697}, \underbrace{10000001}_{1}$

$\underbrace{10000011}_{3}, \underbrace{10000011}_{3}, \underbrace{10000010}_{2}, \underbrace{10000010}_{2}, \underbrace{0000111\ 11100111}_{999}$

$\underbrace{10000101}_{5}, \underbrace{10000001}_{1}, \underbrace{00000011\ 11110100}_{500}$

$\underbrace{00000100\ 00011011\ 10001000}_{69000}, \underbrace{10000111}_{7}$

תוכן קובץ רשימת התפוצה בהקסא:

83 88 05 B9 81 83 83 82 82 0F E7

85 81 03 F4 04 1B 88 87

בנוסף לקובץ רשימות התפוצה יש לעדכן את קובץ המילון שיכיל בטבלה עבור כל מילה את המצביע לרשימת התפוצה. בדוגמה:

רשימת התפוצה של מילה ab מתחילה במיקום 0

רשימת התפוצה של המילה abc מתחילה במיקום 5

רשימת התפוצה של המילה ba מתחילה במיקום 9