בס"ד

**ניתוח סיבוכיות עבור תרגיל א’ סעיף א’**

* **split\_log\_file (filename, chunk\_size=100000)**

**n**- מספר השורות בקובץ

**l**- גודל ה chuksize

**סיבוכיות זמן: O(n**)

הפונקציה עוברת על כל השורות פעם אחת וכותבת אותם בחלקים לקבצים

**סיבוכיות מקום: O(n)**

הפונקציה כותבת את כל השורות לקבצים נפרדים. כל chuksize שורות בקובץ. יש l n\ קבצים סה"כ: O(n\l).

* **count\_errors\_in\_chunk (chunk\_filename)**

**l**-מספר שורות בקובץ (l<n)

**k**- מספר קודי השגיאה השונים שנמצאו בקובץ

**סיבוכיות זמן: O(l)**

הפונקציה עוברת פעם אחת על כל השורות בקובץ

**סיבוכיות מקום: O(1)**

הפונקציה שומרת מילון Counter בגודל k. גודל המילון המקסימלי הוא מספר השגיאות האפשריות וזה מספר קבוע=> O(k)<O(1) סה"כ: O(1)

* **merge\_error\_counts(part\_files)**

**סיבוכיות זמן: O(n)**

הפונקציה עוברת על כל קובץ ומפעילה עליו את פונקציה count\_errors\_in\_chunk סה"כ הפונקציה עוברת על כל השורות בכל הקבצים= O(n) \*O(l)(n\l)O

**סיבוכיות מקום: O(1)**

הפונקציה שומרת בCounter אחד מספר כולל של כל השגיאות וכפי שציינו קודם מספר השגיאות המקסימלי שניתן הוא מספר קבוע ולכן Counter יש גודל קבוע סה"כ: O(1)

* **get\_top\_n\_errors(error\_counts, n)**

**סיבוכיות זמן: O(1)**

הפונקציה בונה ערמת מקסימום בגודל n מתוך K קודי שגיאה שונים O(klogn) k>n. כפי שציינו בפונקציות הקודמות: k מספר קבוע ולכן הערמה בגודל קבוע- O(1)

**סיבוכיות מקום: O(1)**

הפונקציה מחזירה את n הערכים של ערמת המקסימום. N<K סה"כ O(1)

* **פונקציה הmain**

n- מספר השורות בקובץ

l- מספר השורות בקובץ המחולק (גודל ה chuksize)

**סיבוכיות זמן: O(n)**

* קריאה ל split\_log\_file- O(n)
* מציאת הקבצים – O(n/l)
* קריאה ל merge\_error\_count- O(n)
* קריאה ל get\_top\_n\_error- O(1)
* הדפסת הערמה- O(1)

סה"כ: O(n)

**סיבוכיות מקום: O(1)**

* קריאה ל split\_log\_file- O(n)
* מציאת הקבצים – O(n/l) מערך בגודל n\l של שמות הקבצים
* קריאה ל merge\_error\_count- O(1)
* קריאה ל get\_top\_n\_error- O(1)
* הדפסת הערמה- O(1)

סה"כ: O(n)