**5. הסיבוכיות של הפתרון:**

**סיבוכיות זמן:**

* פונקציית הפיצול split\_log\_file פותחת את קובץ הלוג, שזה דורש זמן לינארי ביחס לגודל הקובץ, כלומר סיבוכיות O(L), כאשר L הוא מספר השורות בקובץ.
* פונקציית count\_errors\_in\_chunk סורקת כל שורה של כל חלק, ולכן גם זמן הריצה שלה הוא לינארי ביחס לגודל החלק, כלומר O(C), כאשר C הוא מספר השורות בחלק.
* פונקציית merge\_counters אוספת את כל הספירות מכל החלקים ומעדכנת אותן. זמן הריצה שלה יהיה O(E), כאשר E הוא מספר השגיאות הכולל.
* פונקציית get\_top\_n\_errors אוספת את השגיאות השכיחות ביותר ומחזירה את N השגיאות הנפוצות ביותר. זמן הריצה שלה יהיה O(E log N) אם נניח שהתוצאה צריכה להיות מסודרת.

לכן, סיבוכיות הזמן הכוללת היא O(L + C + E log N), וזה יעיל יחסית למטרות שלך.

**סיבוכיות מקום:**

* השימוש בזיכרון תלוי בכמות החלקים שאתה יוצר בקובץ, כלומר O(C) עבור כל חלק ובסופו של דבר O(C \* k) אם יש לך k חלקים.
* ה-Counter תופס מקום O(E), כלומר המקום שבו נשמרות הספירות של השגיאות.
* סך כל הזיכרון תלוי בכמות החלקים והנתונים.