

## שאלה 1

ד

t  
th  
thr  
thro  
throu  
throug  
through  
thought  
throughth  
throughthe  
throughthel  
thelooking-g  
thelooking-gl  
thelooking-gla  
thelooking-glas  
thelooking-glass  
lewiscarrollchapt  
lewiscarrollchapte  
lewiscarrollchapter  
lewiscarrollchapteri  
lewiscarrollchapteri.  
hadquitealongargumentw  
hadquitealongargumentwi  
hadquitealongargumentwit  
hadquitealongargumentwith  
,andshewasquitepleasedtofi  
,andshewasquitepleasedtofin  
,andshewasquitepleasedtofind  
,andshewasquitepleasedtofindt  
,andshewasquitepleasedtofindth  
,andshewasquitepleasedtofindtha  
,andshewasquitepleasedtofindthat  
inonehandandapieceofbread-and-but  
inonehandandapieceofbread-and-butt  
inonehandandapieceofbread-and-butte  
inonehandandapieceofbread-and-butter  
inonehandandapieceofbread-and-butteri  
inonehandandapieceofbread-and-butterin  
inonehandandapieceofbread-and-butterint  
inonehandandapieceofbread-and-butterinth  
inonehandandapieceofbread-and-butterinthe  
inonehandandapieceofbread-and-butterintheo

inonehandandapieceofbread-and-butterintheot  
 inonehandandapieceofbread-and-butterintheoth  
 inonehandandapieceofbread-and-butterintheothe  
 inonehandandapieceofbread-and-butterintheother  
 inonehandandapieceofbread-and-butterintheother.

## שאלה 2

א

המחרוזת  $abcd$ . הפלט שיתקבל בשתי ההרצות יהיה  $[a', b', c', d']$

ב

```
>>> inter_to_bin(lz77_compress_new('abcdcbcddeeeabcdeee'))
'01100001011000100110001101100100011000111000000000100000110',
+ '1100101011001010110010101100001100000000011100110'

>>> len(-)
108

>>> inter_to_bin(lz77_compress2('abcdcbcddeeeabcdeee'))
'01100001011000100110001101100100011000111000000000100000110',
+
'11001010110010101100101100000000101100100100000000011100011'

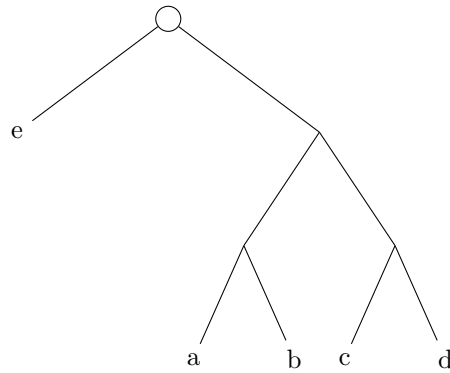
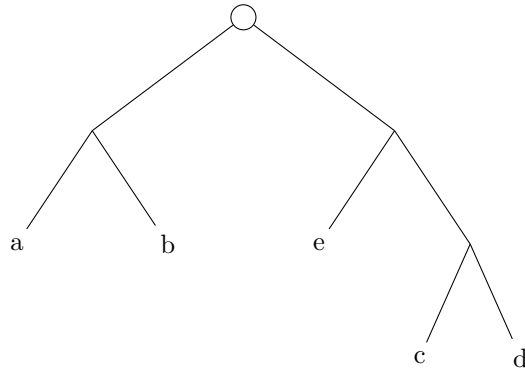
>>> len(-)
118
```

ג

לא קיימת מחרוזת כזו. הסיבה לכך היא שעבור החזרה של  $res2$  בכל פעם ש  $k < 3$   $lz77\_compress\_new$  מחזירה תוצאה זהה לזאת של הפונקציה  $lz77\_compress2$  הבוחרת להציג חזרות באופן חמדני (כל פעם שהיא מוצאת חזרה). באינדוקציה, ניתן להראות כי ערך החזרה של  $lz77\_compress\_new$  הוא זה שהמחרוזת הבינארית המתאימה לו היא המינימלית. לכן יוחזר ערך שונה מזה של  $lz77\_compress2$  תחזיר אם ורק אם המחרוזת הבינארית המתאימה לו קטנה יותר.

### שאלה 3

א



ב

קורפוס לדוגמא:

$$a_1 a_2 a_3 a_4 a_4 a_5 a_5 a_5 = a_1 \cdot a_2 \cdot a_3 \cdot a_4^2 \cdot a_5^3$$

ג

אמיר צודק. לדוגמא עבור הקורפוס:

$$a_1 a_1 a_2 a_2 a_2 a_3 a_3 a_3 a_3 a_4 a_4 a_4 a_4 = a_1^2 \cdot a_2^3 \cdot a_3^4 \cdot a_4^5$$

### שאלה 4

א

המרחק של הקוד הוא 2.

## ב

תמיד ניתן לגלות כי נפלה טעות כלשהי.  
ידוע שנפלה רק טעות אחת. טעות באזור האינפורמציה תגרום להעלאת או הורדת מספר האפסים בו באחד ואז לא תהיה התאמה בינו לבין הערך באזור הביקורת. טעות באזור הביקורת תגרום לכך שמספר האפסים שאמורים להימצא יעלה או ירד בלפחות 1 ואז לא תהיה התאמה בינו לבין איזור האינפורמציה.

לפעמים ניתן לתקן את השגיאות ולפעמים לא.  
דוגמה שניתן לתקן:

$$1000 \cdot 0000 \cdot 1000 \Rightarrow 0000 \cdot 0000 \cdot 10000$$

ישנו רק ביט אחד שצריך לתקן. כעת לפי אזור הביקורת אמורים להיות 8 אפסים אך באזור האינפורמציה ישנם 7 אפסים. אם נשנה את הביט 1 באזור הביקורת נקבל שאמורים להיות 0 אפסים וזו טעות. אם נשנה אחד מהביטים האחרים באזור הביקורת מ0 ל1 נקבל שאמורים להיות יותר מ8 אפסים וזו גם טעות. אם נשנה אחד מהאפסים באזור האינפורמציה ל1 נקבל 6 אפסים ללא התאמה לאזור הביקורת. רק אם נשנה את הביט 1 באזור האינפורמציה ל0 נקבל התאמה בין אזור הביקורת לאזור האינפורמציה.

דוגמה שלא ניתן לתקן:

$$1111 \cdot 1111 \cdot 0001$$

שני מקורות אפשריים:

$$1111 \cdot 1110 \cdot 0001$$

$$1111 \cdot 1111 \cdot 0000$$

## ג

לפעמים ניתן לגלות כי נפלה טעות ולפעמים לא.  
דוגמה בה ניתן לגלות כי נפלה טעות:

$$1111 \cdot 1100 \cdot 0000$$

דוגמה בה לא ניתן לגלות כי נפלה טעות:

$$1111 \cdot 1000 \cdot 0011$$

זוהי מחרוזת חוקית, אך יכלה להיות מחרוזת מקורית עם אינפורמציה שונה, לדוגמה:

$$1111 \cdot 1100 \cdot 0010$$

לפעמים ניתן לתקן את השגיאות ולפעמים לא.  
דוגמה שניתן לתקן:

$$0000 \cdot 0011 \cdot 1000 \Rightarrow 0000 \cdot 0000 \cdot 1000$$

ישנם שני ביטים שצריך לתקן. כל שינוי של ביט באזור האינפורמציה מוסיף או מוריד 0 אחד. שינוי של ביט אחד בלבד באזור הביקורת יאפשר לנו לקבל שם 0 אך אז שינוי של ביט יחד באזור האינפורמציה לא יספיק כדי ליצור ערך חוקי. ניתן לקבל באזור הביקורת גם ערכים גדולים מ-8 אך אלו אינם חוקיים. שינוי של שני ביטים באזור הביקורת יאפשר לנו לקבל שם את הערכים 0, 1, 2, 4 על ידי מחיקת ה-1 ושינוי אחד האפסים ל-1. אך אז בצד השני יהיו 6 אפסים ונקבל מחרוזת לא חוקית. אם לא נשנה את אזור הביקורת נקבל שצריכים להיות 8 אפסים. כעת ישנם 6 אפסים ולכן רק שינוי של שני ה-1 ל-0 יגרום לכך שיהיו 8 אפסים.

דוגמה שלא ניתן לתקן:

$$1111 \cdot 1100 \cdot 0000$$

שני מקורות אפשריים:

$$1111 \cdot 1111 \cdot 0000$$

$$1111 \cdot 1110 \cdot 0001$$

ד

תמיד ניתן לגלות כי נפלה טעות כלשהי.  
טעויות שנופלות באזור הביקורת מעלות את ערכו ואז ישתמע ממנו שאמורים להיות יותר אפסים משהיו באמת. טעויות שנופלות באזור האינפורמציה מורידות את מספר האפסים. לכן אף אזור לא יכול לחפות על טעות באזור השני ומכאן שניתן לגלות כל טעות.

לפעמים ניתן לתקן את השגיאות ולפעמים לא.  
דוגמה שניתן לתקן:

$$0000 \cdot 0001 \cdot 0110 \Rightarrow 0000 \cdot 0001 \cdot 0111$$

באזור הביקורת הסיבית הרביעי (מימין) הוא אפס ולכן היו פחות מ-8 אפסים במקור. באזור האינפורמציה ישנם 7 אפסים שבהכרח הופיעו גם במקור, לכן הספרה הימנית של אזור האינפורמציה חייבת להיות 1 במקור. כעת נותר לתקן את אזור הביקורת כך שיכיל את הערך 7.

דוגמה שלא ניתן לתקן:

1111 · 1111 · 0001

שני מקורות אפשריים:

1111 · 1111 · 0000

1111 · 1110 · 0001