

# מושגים בסיסיים

- סימונים עבור קוד

א"ב של קובץ המקור.	↔	$S = \{s_1, \dots, s_n\}$	○
הסתברויות של ה-א"ב.	↔	$P = \{p_1, \dots, p_n\}$	○
מילות הקוד.	↔	$C = \{c_1, \dots, c_n\}$	○
אורכי מילות הקוד (כמות הסיביות).	↔	$ C  = \{ c_1 , \dots,  c_n \}$	○

- UD (Uniquely Decipherable) – פענוח קוד בצורה יחידה.

$Prefix - Free \Rightarrow UD$  ○  
 $UD \not\Rightarrow Prefix - Free$  ○

○ אם יש קוד UD אך לא פרפיקסי נוכל להפוך את מילות הקוד ונקבל קוד פרפיקסי ובהכרח UD ומייד.

- מחרוזת אינסופית למחצה – מחרוזת אינסופית מכיוון אחד.

○ התחלת הקוד ברורה אך ההמשך יכול להיות אינסופי.

- Complete Code – קוד שלם.

○ כל מחרוזת אינסופית למחצה ניתנת לפענוח בצורת UD.  
○ קוד שלם יוצר עץ מלא.

- Prefix-Free CodeWords – קוד חסר רישות \ קוד פרפיקסי.

○ אף מילת קוד אינה רישא של מילת קוד אחרת.

- Instantaneous Code – קוד מידי.

○ ברגע שמילת הקוד מסתיימת, המפענח יודע את הפענוח של מילת הקוד.

- Minimum Redundancy Code – קוד בעל יתרות מינימאלית (קוד אופטימאלי).

○ בהינתן הסתברויות או התפלגות מסוימת, אין עוד אפשרות של קידודים אחרים שיכולים להפחית את אורך מילת הקוד הממוצעת ( $E(C, P)$ ) מעבר למה שנקבל בקוד בעל היתרות המינימאלית.  
○ קוד אופטימאלי יוצר עץ מלא.



# הגדרות ונוסחאות

- Expected Codeword Length – אורך ממוצע של מילות הקוד.

$$E(C, P) = \sum_{i=1}^n p_i \cdot |c_i|$$

- אינפורמציה – כמות הביטים המינימלית לייצוג  $s_i$ .

$$I(s_i) = -\log_2(p_i)$$

- אם  $p_i = 1$  אז  $I(s_i) = 0$
- אם  $p_i = 0$  אז  $I(s_i) = 1$
- ככל שההסתברות נמוכה יותר נצטרך להעביר יותר מידע.
- ניתן לבדוק את כמות הביטים המינימלית לייצוג מחרוזת על ידי סכימת האינפורמציה של כל התווים שלה.

- אנטרופיה – סכום משוכלל של כמות האינפורמציה לייצוג מחרוזת.

$$H(P) = \sum_{i=1}^n p_i \cdot I(s_i)$$

- בהינתן התפלגות  $P$ , נגדיר את האנטרופיה כסכום משוכלל של כמויות האינפורמציה.
- האנטרופיה משמשת כחסם תחתון לאורך הממוצע של מילות הקוד.

- Kraft – אי שיוויון קראפט.

$$K(C) = \sum_{i=1}^n 2^{-|c_i|}$$

- $K(C) \leq 1$      $\Leftrightarrow$     קיים קוד  $C'$  כך ש:
  - $E(C, P) = E(C', P)$     ■
  - $|C| = |C'|$     ■
  - $C'$  הוא קוד פרפיקסי! (בהכרח מיידי).

- $K(C) = 1$      $\Leftrightarrow$      $C$  הוא קוד שלם.

- $K(C) \leq 1$      $\Leftrightarrow$

# שיטות מידול

- שיטת מידול סטטית

בשיטה זו לא משנה איזה קובץ יש לנו , מניחים שכל אחד מהתווים מופיע בהסתברות  $\frac{1}{256}$  ,  
לכן משתמשים בקוד ascii לקידוד.  
במקרה זה אין Prelude.

- שיטת מידול סטטית למחצה

בשיטה זו עוברים על הקובץ, סופרים כמה תווים יש לנו ואז מניחים שההסתברות ביניהם אחידה.  
בשיטה זו יש Prelude הבנוי בצורה הבאה:  
– 8 ביטים כדי להעביר את המספר  $n$  המסמל את גודל הא"ב.  
–  $n * 8$  על מנת להעביר את התווים עצמם (נשתמש ב-ascii לכן כל תו הוא 8 ביטים).  
– מילות הקוד עצמם.

- שיטת מידול סטטית למחצה

בשיטה זו נעבור על הקובץ, סופרים כמה תווים יש לנו וגם מה פעמים מופיע כל תו כדי לחשב את ההסתברות.  
ההסתברות של תו תהיה  $p_i = \frac{\text{מספר המופעים}}{\text{גודל ההודעה}}$   
ה-Prelude בנוי בצורה הבאה:  
– 8 ביטים כדי להעביר את המספר  $n$  המסמל את גודל הא"ב.  
–  $n * 8$  על מנת להעביר את התווים עצמם (נשתמש ב-ascii לכן כל תו הוא 8 ביטים).  
–  $n * 8$  על מנת להעביר את השכיחות לכל תו.

- מודל מרקוב מסדר ראשון

במודל זה מניחים תלות בין התווים. אם אנו יודעים שתו מסוים מופיע ואחריו יש לנו הסתברות גבוהה יותר לקודד תווים מסוימים נעלה את ההסתברות של אותו תו, לכן כמות האינפורמציה שלו תרד וכך גם האנטרופיה.  
במודל זה הא"ב שלנו יהיה כל הזוגות האפשריים מהא"ב המקורי של התווים הבודדים.

- ויזואליזציה מבוססת עץ

עבור קוד פרפיקסי ניתן לבנות עץ כך שהמילים יהיו בעלים.  
מכיוון שזהו קוד חסר רישות, יהיה מסלול יחיד מהשורש עד לעלים.  
הקידוד והפענוח יהיה על ידי מעבר על העץ (= איטי).

- עץ אופטימלי – עץ השייך לקוד אופטימלי (בעל יתירות מינימאלית).

בעץ אופטימלי ההסתברויות הנמוכות ביותר נמצאות ברמה הנמוכה ביותר.

- עץ D-ary מלא

לכל צומת יש D בנים בדיוק.

לעץ D-ary מלא עם k צמתים פנימיים, יש  $k + 1 * (D - 1)$  עלים.

1) נתון ה-א"ב  $\{a, b, c\}$  ונתונה המחרוזת הבאה:  $ababbac$

א) חשב את מספר הסיביות הדרוש לתיאור המודל (Prelude) במודל סטטי למחצה (Semi-static), בו מניחים התפלגות אחידה של התווים.

פתרון: 8 ביטים לתיאור גודל ה-א"ב.  
 $3 \cdot 8$  להעברת האותיות עצמן (כל char בית אחד – 8 ביטים).

$$\text{סה"כ } 8+3 \cdot 8=32$$

ב) חשב את מספר הסיביות הדרוש לתיאור המודל (Prelude) במודל סטטי למחצה עם הסתברויות עצמאיות (Semi-static self probabilities). הנח שמספר הסיביות הדרוש לקידוד הסתברות אחת הוא 4.

פתרון: 8 ביטים לתיאור גודל ה-א"ב.  
 $3 \cdot 8$  להעברת האותיות עצמן (כל char בית אחד – 8 ביטים).  
 $3 \cdot 4$  ביטים להעברת ההסתברויות עצמן (לפי הסדר, אין צורך במידע נוסף להתאמה).

$$\text{סה"כ } 8+3 \cdot 8+3 \cdot 4=44$$

2) הוכח או הפרך:

נתונים אורכי מילות הקוד  $\{1,2,3,3,4\}$   
לא קיים קוד חסר רישות (קוד פרפיקסי) בעל אורכים כ"ל.

$$\sum_{i=1}^n 2^{-l_i} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} + 2^{-4} = \frac{17}{16} > 1$$

נבדוק את האורכים בעזרת קראפט:  
מכיוון שהקראפט גדול מ-1,  
לא קיים קוד חסר רישות עבור אורכים אלו.

3) נתון הא"ב  $\Sigma = \{a, b\}$  בהסתברויות  $P = \{0.8, 0.2\}$  בהתאמה, ונתונה המחרוזת הבאה:  $aaab$ .  
חשב מבחינה תיאורטית את מספר הסיביות המינימאלי הנחוץ לקידוד המחרוזת (האנטרופיה).

$$H(P) = -[0.8 \cdot \log_2 0.8 + 0.2 \cdot \log_2 0.2] = 0.7219$$

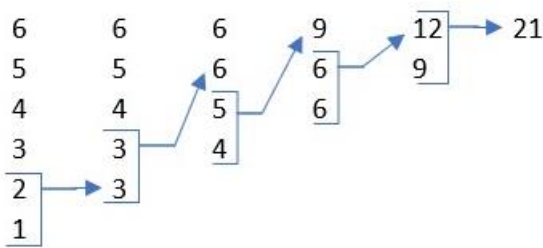
4) הוכח או הפרך:

הקוד הבא הינו אופטימאלי

קוד	שכיחות	תו
000	1	a
001	3	b
100	2	c
101	4	d
11	5	e
01	6	f

נחשב את אורך מילת הקוד הממוצעת עבור קוד זה

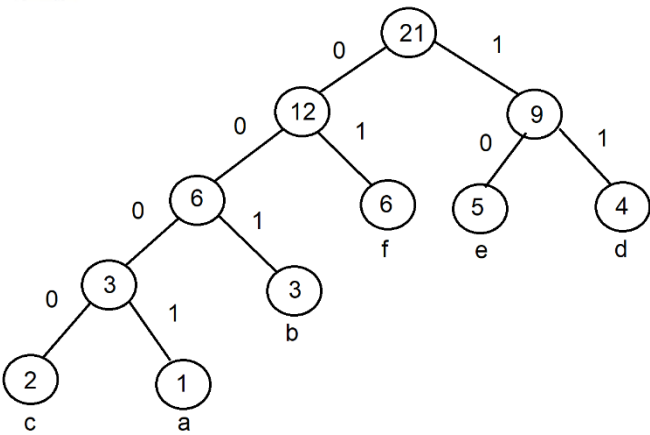
$$E(C,P) = \frac{1}{21} * 3 + \frac{3}{21} * 3 + \frac{2}{21} * 3 + \frac{4}{21} * 3 + \frac{5}{21} * 2 + \frac{6}{21} * 2 = \frac{52}{21} = 2.476$$



כעת, נבנה קוד הפמן  $C_h$  עבור שכיחויות אלה

נצייר את העץ על מנת ליצור את מילות הקוד

נתבונן בקודים המתקבלים



תו	קוד
a	0001
b	001
c	0000
d	11
e	10
f	01

נחשב את ה  $E(C_h,P)$

$$E(C_h,P) = \frac{1}{21} * 4 + \frac{2}{21} * 4 + \frac{3}{21} * 3 + \frac{6}{21} * 2 + \frac{5}{21} * 2 + \frac{4}{21} * 2 = \frac{17}{7} = 2.428$$

מכיוון שקוד הפמן יוצר קוד אופטימאלי, נשווה בין  $E(C,P)$  ל-  $E(C_h,P)$ . נשים לב כי על ידי האפמן קיבלנו קוד טוב יותר, לכן הטענה אינה נכונה והקוד אינו אופטימאלי.

5) נתונות מילות הקוד הבאות :  $a = 101, b = 1101, c = 1011, d = 1100$   
האם קוד זה ניתן לפיענוח חד ערכי (UD)?

פיתרון 1: מבחן UD.

שלב 1: {101,1101,1011,1100 | 1}  
שלב 2: {101,1101,1011,1100 | 1,01,101,011,100}

מצאנו מילה קיימת ולכן הקוד אינו ניתן לפענוח חד ערכי.

בנוסף, ניתן לראות כי 1011101 יכול להתפרש גם כ-ca וגם כ-ab.

(6) נתון א"ב בגודל  $x + 3$ . יש לבנות קוד  $C = \{c_1, \dots, c_{x+3}\}$  הניתן לפענוח בצורה יחידה (UD), כך ששלושת מילות הקוד הראשונות -  $c_1, c_2, c_3$  מכילות 3 סיביות ושאר מילות הקוד  $c_4, \dots, c_{x+3}$  מכילות 8 סיביות.

**חשב מהו גודל הא"ב המקסימאלי כך שמצב זה אפשרי.**

א"י שיויוין קראפט מוודא שעבור כל מילה יש מסלול יחיד משורש העץ לעלה בעומק השווה לאורך המילה ושתי מילים המשויכות לאותו צומת או לצאצא שלו.

$$K(C) = \sum_{i=1}^{x+3} 2^{-l_i} = 3 \cdot 2^{-3} + x \cdot 2^{-8} \leq 1 \quad \text{לכן, צריך להתקיים}$$

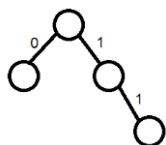
$$\frac{3}{8} + \frac{x}{2^8} \leq 1 \Rightarrow \frac{x}{256} \leq \frac{5}{8} \Rightarrow 8x \leq 1280 \Rightarrow x \leq 160$$

(7) הוכח או הפרך: קוד מידי הוא קוד שלם.

לא נכון. נראה שתי פתרונות:

פתרון 1: קוד אונארי הוא קוד מייד'י אך לא קוד שלם.

בקוד אונארי אנו יודעים כי המילה מסתיימת ב-0, לכן בפיענוח הקוד נדע בדיוק עם סיום המילה מהי המילה. בניגוד לכך שאונארי קוד מיידי הוא לא קוד שלם. מכיוון שהעץ שהוא יוצר אינו עץ מלא.



פתרון 2: נבנה קוד מידיי אך לא שלם.

ניקח את מילות הקוד  $\{0, 11\}$  ובבנה את העץ –  
 ניתן לראות כי זהו קוד פרפיקסי (לכן בהכרח מידי!),  
 אך לא קוד שלם מכיוון שהעץ הנוצר ממנו הוא לא עץ מלא.

## Unary

- מילת הקוד  $x_i = '1^{i-1}0'$
- אם ה-א"ב שלנו הוא סופי נוכל לחסוך במילה האחרונה, להוריד את ה-0 ולקבל עץ מלא ואז  $Kraft(X) = 1$ .
- בקוד אונרי ה-0 מפריד לנו בין מילה למילה, אך מכיוון שנדע כמה 1'ים יש לנו במילה האחרונה שהורדנו ממנה את ה-0 נוכל להבין כי זו המילה היחידה עם כמות ה-1'ים הזו.
- מילה אחת יכולה להופיע המון פעמים בקובץ, לכן ברגע שחסכנו סיבית אחת חסכנו סיבית מכל מופע שלה.
- נקבל קוד אופטימאלי כאשר ההסתברויות הן בחזקות של  $\frac{1}{2}$ .
- כך כמות האינפורמציה למילה שווה בדיוק לאורך מילת הקוד שלה.
- ההבדל באופטימאליות של קוד סופי לאינסופי כאשר ההסתברויות הן בחזקות של  $\frac{1}{2}$  הוא שב-א"ב סופי התו האחרון יהיה בהסתברות של  $\left(\frac{1}{2}\right)^{n-1}$ , כך שסכום ההסתברויות יהיה שווה ל-1.

### Simple Binary Code

- אנו מעוניינים בקוד עם אורך קבוע, עבור  $n$  תווים ניתן מילת קוד באורך הערך העליון של  $\log_2 n$ .
  - כאשר מספר התווים  $n$  שלנו הוא חזקה של 2 ( $n = 2^k$ ), ניתן מילות קוד באורך  $k$  מכיוון ש  $\log_2 2^k = k$ .
  - במקרה כזה אנו עלולים לקבל קוד שיוצר עץ לא מלא.
- דוגמא: עבור  $n = 7$  אנחנו צריכים מילות קוד באורך  $\log_2 7 = 2.8 \dots$  (נעגל למעלה ונקבל 3) נשים לב שבעץ שמתקבל מהקוד, עבור הצומת '11' יש לנו רק בן יחיד 110, אבל 111 לא קיים ולכן העץ לא מלא.

### Minimal Binary Code

- דוגמא: עבור  $n = 6$

100	00
101	01
110	
111	

דוגמא: עבור  $n = 5$

110	00
111	01
	10

## שאלות

(1) נניח ש- $n$  הוא חזקה של 2. מתי הקוד הבינארי המינימאלי הוא קוד אופטימאלי?

**פיתרון:** הציפייה היא שכמותה אינפורמציה תהיה שווה לאורך מספר הסיביות עבור מילת קוד.

Golomb

גולומב מתייחס ל"דלי" בגודל קבוע, בניגוד לקוד אליאס, שה"דליים" גדלו בצורה אקספוננציאלית.

$0 \leq r \leq b$

$\uparrow$

$\text{Golomb\_encode}(x, b) \{$

$q \leftarrow (x-1) \text{ div } b;$   $\leftarrow$  החלק השלם

$r \leftarrow x - q \cdot b;$   $\leftarrow$  השארית

$\text{Unary\_encode}(q+1);$   $\Rightarrow$  החלק הראשון של מילה הקוד

$\text{Minimal\_binary\_encode}(r, b);$   $\Rightarrow$  החלק השני של מילה הקוד

$\leftarrow$  מקודדים סטנדרטיים  
אל השארית

$\leftarrow$  מקודדים ב-MB  
אל השארית

$\leftarrow$  מילה הקוד ה-r

$\leftarrow$  מילה הקוד ה-r

אלגוריתם קידוד:

x - האינוקס של מילה הקוד  
שחזים לקוד [האינוקס מתחיל מ-1]

b - האופס הקבוע של הדל'

דוגמא: נרצה לקודד את  $x = 9$  עבור דלי בגודל  $b = 5$ .

- $q = \frac{(9-1)}{5} = 1$
- $r = 9 - 1 * 5 = 4$
- $\text{Unary}(1 + 1) = 10$
- $\text{MBE}(4,5)$
- $\log_2 5 = 2.32 \dots$  נעגל למעלה = 3 אורך קבוצת המילים הארוכות
- $\log_2 5 = 2.32 \dots$  נעגל למטה = 2 אורך קבוצת המילים הקצרות
- $2^3 - 5 = 3$  מספר המילים הקצרות יהיה
- $2 * 5 - 2^3 = 2$  מספר המילים הארוכות יהיה

111 (5 110 (4 10 (3 01 (2 00 (1



$\text{Golomb\_decode}(b) \{$  הזרוע של הקוד  
 $q \leftarrow \text{Unary\_decode}() - 1;$  משיב (-1), משיב  
 $r \leftarrow \text{Minimal\_binary\_decode}(b);$  מקדוב עשירי  $(q+1)$   
 $\text{return } r + q \cdot b;$   
 $\}$

דוגמא: עבור הקוד 10110.

נשים לב כי הדלי בגודל  $b = 5$  והחלק האונרי שלנו הוא 10.

- $q = \text{DeUnary}(10) - 1 = 1$
- $r = (5, 110) = 4$  זה בעצם האינדקס של מילת הקוד 110 בא"ב בגודל 5 ב-MBE.
- $4 + 1 * 5 = 9$

רייס הוא מקרה פרטי של גולומב עבור דליים בגודל  $b = 2^k$  (חזקה של 2).

אלגוריתם קידוד: נרצה לקודד את  $x$  עבור דלי בגודל  $b = 2^k$

- עבור החלק האונארי –
  - נייצג בבינארית את  $x - 1$ .
  - נמחק את  $k$  הביטים הימניים (נשמור אותם בצד).
  - נייצג באונרי את הערך של הייצוג הבינארי (לאחר שמחקנו את הביטים)  $+ 1$ .
- עבור החלק הבינארי –
  - נוסיף את אותם ביטים שמחקנו.

דוגמא: נרצה לקודד את  $x = 9$  עבור דלי בגודל  $b = 4$ .  
נשים לב ש  $k = 2$ .

- עבור החלק האונרי –
  - $\text{Binary}(8) = 1000$
  - $\text{Shifting2Bits}(1000) = 10$
  - $\text{BinaryToDecimal}(10) = 2$
  - $\text{Unary}(2 + 1) = 110$
- עבור החלק הבינארי –
  - הביטים שנמחקו – 00.

- לכן הקוד הוא 11000

קוד אליאס משלב בין קוד בינארי ואונרי.  
יש שני סוגים של קוד אליאס -  $C_\gamma$  (גאמה) ו- $C_\delta$  (דלתא).

קידוד  $C_\gamma$ :

- בחלק הראשון נכתוב בקוד אונארי את מספר הסיביות הנדרשות לייצוג  $x$  בבינארי.
  - מס' הסיביות של  $x$  יהיה הערך התחתון של  $1 + \log_2 x$
- בחלק השני נכתוב את הייצוג הבינארי של  $x$  ללא ה-1 מוביל.
  - לכן מס' הסיביות של הייצוג הבינארי ללא 1 מוביל הוא הערך התחתון של  $\log_2 x$ .
- סה"כ עבור שני החלקים מקבלים  $|x_\gamma| = 1 + 2 * \log_2 x$  (בערך תחתון).

דוגמא לקידוד המספר 12	דוגמא לקידוד המספר 25
12 בבינארי $\Leftarrow$ 1100 (4 סיביות) 4 באונרי $\Leftarrow$ 1110	25 בבינארי $\Leftarrow$ 11001 (5 סיביות) 5 באונרי $\Leftarrow$ 11110
1110100	111101001

פענוח  $C_\gamma$ :

- נקרא את הקוד משמאל לימין עד ה-0 הראשון.
- נפרש את הקוד האונרי לספרה עשרונית (x).
- נקרא את  $x - 1$  הביטים הבאים, נוסף להם 1 מוביל ונמיר מבינארי לעשרוני.

קידוד  $C_8$ :

- בחלק הראשון נכתוב ב- $C_7$  את מספר הביטים הנדרשים כדי לייצג את x בבינארי.
- בחלק השני נכתוב את הייצוג הבינארי של x ללא ה-1 המוביל.

דוגמא לקידוד המספר 13	דוגמא לקידוד המספר 30
13 בבינארי $\Leftarrow$ 1101 (4 סיביות)	30 בבינארי $\Leftarrow$ 11110 (5 סיביות)
4 ב- $C_7$ :	5 ב- $C_7$ :
4 בבינארי $\Leftarrow$ 100 (3 סיביות)	5 בבינארי $\Leftarrow$ 101 (3 סיביות)
3 באונרי $\Leftarrow$ 110	3 באונרי $\Leftarrow$ 110
4 ב- $C_7$ $\Leftarrow$ 11000	5 ב- $C_7$ $\Leftarrow$ 11001
13 ב- $C_8$ $\Leftarrow$ 11000101	30 ב- $C_8$ $\Leftarrow$ 110011110

השוואה בין Elias codes - $C_7$ ו- $C_8$		
x	Elias $C_7$ code	Elias $C_8$ code
1	0	0
2	10 0	100 0
3	10 1	100 1
4	110 00	101 00
5	110 01	101 01
6	110 10	101 10
7	110 11	101 11
8	1110 000	11000 000
9	1110 001	11000 001
⋮	⋮	⋮

שאלות

1) נתונה המחרוזת הבאה של סיביות (משמאל לימין): 0111011111010

האם מחרוזת זו יכולה לייצג קידוד של מספרים בעזרת Elias Code -  $C_\gamma$ ?  
אן כן, פענח את המחרוזת, אחרת, הסבר מדוע זה לא יתכן.

- 0 באונרית = 1  
לכן נקרא 0 סיביות, נוסף 1 מוביל ונקבל 1  
1 בבינארית = 1
- 1110 באונרית = 4  
לכן נקרא 3 סיביות, נוסף 1 מוביל ונקבל 1111  
1111 בבינארית = 15
- 110 באונרית = 3  
לכן נקרא 2 סיביות, נוסף 1 מוביל ונקבל 110  
110 בבינארית = 6

הקוד ניתן לפענוח ולכן הספרות המתקבלות הן 1 15 6

(2) הוכח או הפרך באמצעות דוגמא נגדית:  
כל מספר המקודד בקוד  $C_Y$ , מקודד במספר שווה או גדול יותר של סיביות מאשר בקוד  $C_\delta$ .

המשפט אינו נכון.  $|3_{C_Y}| < |3_{C_\delta}|$ .

(3) נתון ה-א"ב  $\{a, b, c\}$  ונתונה המחזורית הבאה: *ababbac*

(א) השתמש באליאס קוד (סי גאמה) לקידוד המחזורית.

פיתרון: נשייך לכל תו מה-א"ב שלנו קוד מסי גאמה (נתחיל מהקצר ביותר):  $a = 0, b = 100, c = 101$

לכן -  $010001001000101$ .

(ב) חשב את האורך הממוצע של מילת הקוד.

פיתרון:  $3 * 1 * \frac{3}{7} + 3 * 3 * \frac{3}{7} + 1 * 3 * \frac{1}{7} = \frac{39}{7}$

Shannon

בקוד שנון השאיפה הייתה להתקרב לאנטרופיה.

אלגוריתם קידוד:

- נסדר את ההסתברויות בסדרה מונוטונית יורדת.
- נחלק לשתי קבוצות השוות בסכום ההסתברויות שלהם (פחות או יותר).
- לקבוצה השמאלית ניתן את הספרה 0 ולימנית את הספרה 1.
- הקבוצה הימנית תהיה בהכרח גדולה או שווה בגודלה לשמאלית.

Prob.	0.67	0.11	0.07	0.06	0.05	0.04
	0	1				
		0	1			
		0	1	0	1	
				0	1	
Code	0	100	101	110	1110	1111

Huffman

(1) מהו ה-Prelude עבור עץ האפמן?

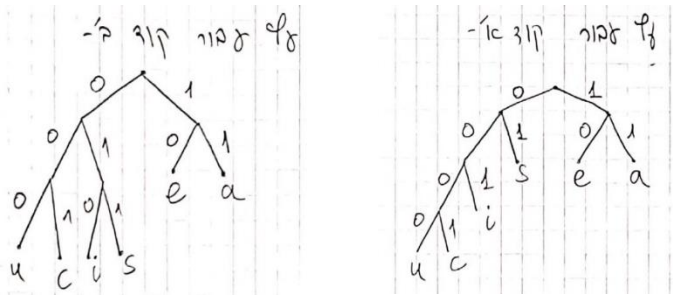
- 8 סיביות עבור גודל ה-א"ב.
- 8\*מספר התווים עבור התווים עצמם
- 8\*8 לשכיחויות
- מספר העלים + מספר העלים – 1

(2) על טבלת התווים והשכיחויות הבאה הופעל אלגוריתם הפמן בשתי צורות כך שהתקבלו שני קודים:

תו	שכיחות	קוד א'	קוד ב'
a	50	11	11
c	15	0001	001
e	45	10	10
i	20	001	010
s	x	01	011

u	10	0000	000
---	----	------	-----

א) שרטט את שני עצי הפמן המתאימים לקודים שהתקבלו.



ב) חשב את x (שכיחות הסימן s) ותן תשובה מספרית מדויקת.

אלגוריתם האפמן עובד בצורה שבה הוא לוקח בכל פעם את ההתסברויות הקטנות ביותר.

שלב 1: קוד א':  $u + c = 10 + 15 = 25$   
קוד ב':  $u + c = 10 + 15 = 25$

שלב 2: קוד א':  $i + 25 = 20 + 25 = 45$   
קוד ב':  $i + s = 20 + x = ?$

מכיוון שבקוד א' אלגוריתם האפמן חיבר את i עם 25, גם בקוד ב' אלגוריתם האפמן יחבר את i עם 25. לכן,  $x = 25$ .

ג) פענח את המילה הבאה (משמאל לימין) לפי קוד ב': 1100100110011011

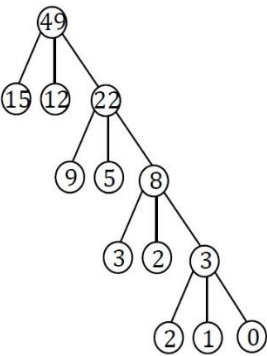
נעבור על העץ עד שנגיע לעלה: access

3) נתונות השכיחויות 1,2,2,3,5,9,12,15 עבור האותיות A, B, C, D, E, F, G, H בטקסט מסוים. ברצוננו לבנות עץ Huffman טרנרי ולהשוות אותו לעץ Huffman בינארי.

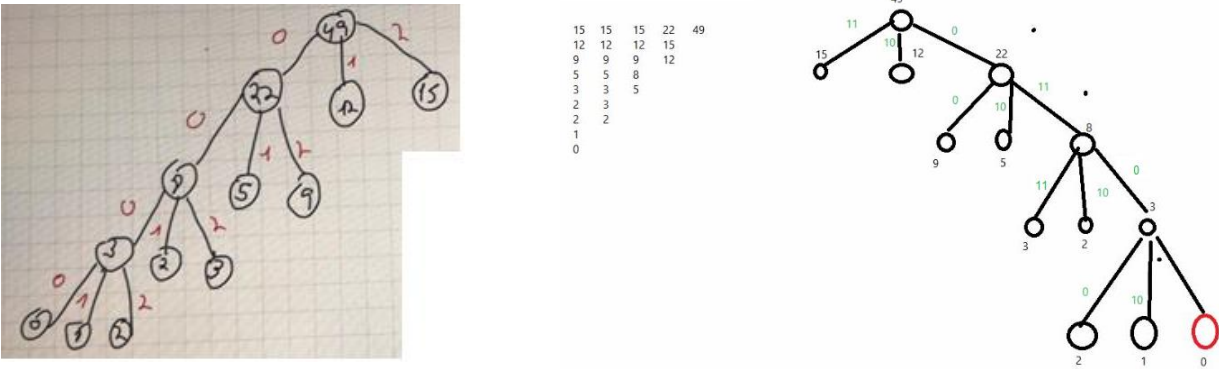
א) בנה עץ האפמן אופטימאלי עבור תדירויות אלו.

נציב בנוסחא למספר העלים  $(D - 1) * k + 1 = (3 - 1) * k + 1 = 2k + 1$  וקיבלנו שמספר העלים הוא אי-זוגי. מכיוון שיש לנו 8 מילים, נוסיף עלה מיותר ונסמן אותו בתדירות 0.

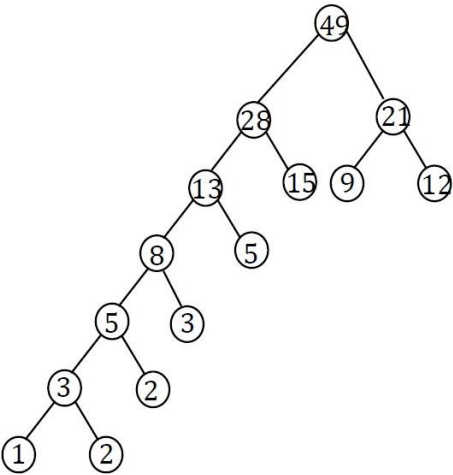
15	15	15	15	49
12	12	12	12	
9	9	9	22	
5	5	5		
3	3	8		
2	2			
2	3			
1				
0				



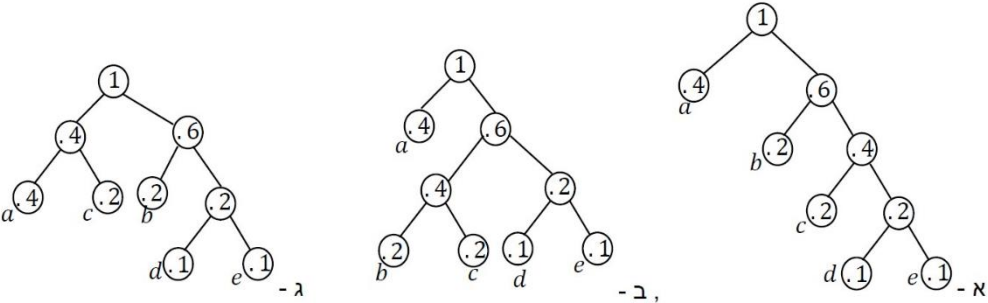
ב) בנה מילות קוד טרנריות עבור האותיות ואחר-כך צור את מילות הקוד הבינאריות על ידי החלפת שלושת התווים הטרנריים על ידי 0,10,11 בהתאמה. דאג לבצע את ההחלפה בצורה הטובה ביותר מבחינת דיחסה.



בנה/י עץ Huffman בינרי וחשבו/י את גדלי הטקסטים הדחוסים בשתי השיטות. איזו שיטה עדיפה?  
פתרון:



4) נתון האב {a, b, c, d, e} והסתברויות {0.4, 0.2, 0.2, 0.1, 0.1} בהתאמה. בעץ הבינרי:  $2 \cdot (12 + 15 + 9) + 3 \cdot 5 + 4 \cdot 3 + 5 \cdot 2 + 6 \cdot 1 = 128$  בעץ הטרינרי:  $6 \cdot (1 + 2) + 5 \cdot 2 + 4 \cdot 3 + 3 \cdot 5 + 2 \cdot (15 + 9 + 12) = 127$   
א) תן שלושה קודי האפמן שונים היכולים להתקבל מהסתברויות אלו.



ב) חשב את האורך הממוצע של כל קוד.

פתרון:  
א -  $0.1 \cdot (4 + 4) + 0.2 \cdot (2 + 3) + 0.4 \cdot 1 = 2.2$   
ב -  $0.1 \cdot (3 + 3) + 0.2 \cdot (3 + 3) + 0.4 \cdot 1 = 2.2$   
ג -  $0.1 \cdot (3 + 3) + 2 \cdot (0.2 + 0.2 + 0.4) = 2.2$

$$-(2 \cdot 0.1 \cdot \log_2 0.1 + 2 \cdot 0.2 \cdot \log_2 0.2 + 0.4 \cdot \log_2 0.4) = 2.122 - \kappa$$

פתרון:

**פתרון:** זה לא משנה. האורך הממוצע של מילות הקוד עבור שלושת הקודים הוא אותו הדבר.

12	12	12	50
11	11	11	
8	8	8	
7	7	19	
4	4		
4	4		
3	4		
1			
0			
0			

ננונה ההודעה הבאה מעל האיב  $\{a, b, c, d, e, f, g, h\}$  שבו האותיות מופיעות עם שכיחות

Canonical Huffman Tree

12, 11, 8, 7, 4, 4, 3, 1 בהתאמה.

- מילות הקוד יבין בסדרה מונוטונית יורדת.
- בנוח קוד הופמן 4 איב לא יבין, ואם מילות הקוד בנויות מהתווים  $A = \{\alpha, \beta, \gamma, \delta\}$  ולא בלוק יקבל מספרים בינאריים עוקבים.

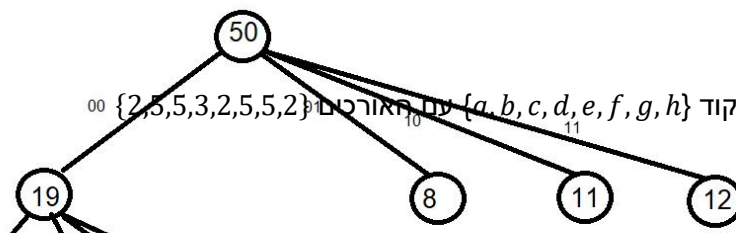
מהסיביות  $\{0, 1\}$  כרגיל. כדי לקבל את הקידוד הבינרי הסופי, התווים ב-  $A$  מתורגמים לבינרית

1) צורה יחידה של הקידוד קנוני? אחד מהקודים  $\{0, 10, 110, 111\}$  או  $\{00, 01, 10, 11\}$ .

מהו הקוד שנוצר? מה מספר סיביות ההודעה הדחוסה בשיטה זו? הסבר!

8 0: ביטוי עבור גודל הא"ב

התו הראשון בבליק השני הגדול ביותר + אורך הקוד שלו באונר' + מילת הקוד עצמה  
 $(E,C) = 6(1/50 + 3/50) + 4(4/50 + 4/50 + 7/50) + 2(8/50 + 11/50 + 12/50) = 2.92$



(2) נקודד את מילות הקוד  $\{a, b, c, d, e, f, g, h\}$  עם פאורכות  $\{2, 5, 5, 3, 2, 5, 5, 2\}$



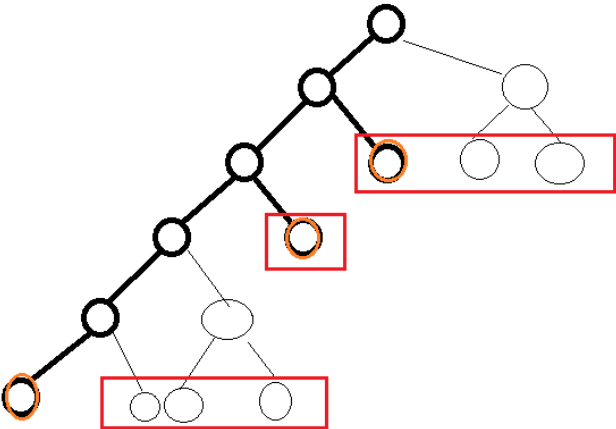
index :	1	2	3	4	5	6	7	8	
char :	a	b	c	d	e	f	g	h	
len :	2	5	5	3	2	5	5	2	
cw :	1	0	1	1	2	2	3	3	$cw[i] = nc[len(i)], nc[len(i)]++$
code:	01	00000	00001	001	10	00010	00011	11	נייצג בבינארית את cw[i] באורך len(i)
num :	0	3	1	0	4				
fc :	2	1	1	2	0				
nc :	2	4	2	2	4				

3) נעביר את הפרילוד

Prelude:  $8 + 8 * 8 + (b + 11110 + 00000 + d + 110 + 001 + a + 10 + 01) = 116$

4) נפענח מהפרילוד (נפענח את num באמצעות ציור העץ לפי מילות הקוד הראשונות בכל רמה).

index :	1	2	3	4	5	6	7	8
char :	b	c	f	g	d	a	e	h
len:	5	5	5	5	3	2	2	2
cw:	0	1	2	3	1	1	2	3
code:	00000	00001	00010	00011	001	01	10	11
i:	1	2	3	4	5			
num:	0	3	1	0	4			
fc:	2	1	1	2	0			
nc:	2	4	2	2	4			



שאלה 3 ממטלה 3:

בהגדרת עץ שלד (Skeleton Huffman Tree), יצרנו תחילה עץ Huffman קנוני ולאחר מכן קיצצנו את העץ, כך שעלה מייצג תת עץ שלם בעץ ה-Huffman הקנוני.

הראה כי ה-Skeleton Huffman Tree שנוצר בצורה כזו אינו תמיד מינימלי מבחינת הצמתיים הנותרים (כלומר, תן דוגמא לעץ שבו דווקא אם העץ אינו מתחיל בעץ קנוני ומקצצים אותו באותה הדרך, נקבל מספר קטן יותר של צמתיים בעץ שמתקבל).

נראה דוגמא עבור האורכים הבאים :

Index (i)	1	2	3	4	5	6	7	8	9	10	11	12
Length(i)	2	3	3	3	4	4	4	4	5	5	5	5

נעבוד לפי אלגוריתם האפמן קנוני ונראה את העץ המתקבל :

(1) מספר העלים בגודל index (i) :

Index (i)	1	2	3	4	5
Num	0	1	3	4	4

(2) הערך הדצימלי של הקוד הבינארי הראשון בבלוק בגודל index (i) :

Index (i)	1	2	3	4	5
Firstcode	2	3	3	2	0

(3) הערך הדצימלי של הקוד הבינארי שנרצה לתת לעלה הבא בבלוק בגודל index (i) :

Index (i)	1	2	3	4	5
Nextcode	2	3 4	3 4 5 6	2 3 4 5 6	0 1 2 3 4

(4) הערך הדצימאלי של הקוד הבינארי עבור עלה index (i) :

Index (i)	1	2	3	4	5	6	7	8	9	10	11	12
Codeword	3	3	4	5	2	3	4	5	0	1	2	3

(5) הקודים הבינאריים בגודל האורכים length (i) לפי הערכים הדצימאליים המתאימים ב – codeword :

Index	Length	CodeWord	BinaryCode
1	2	3	11
2	3	3	011
3	3	4	100
4	3	5	101
5	4	2	0010
6	4	3	0011
7	4	4	0100
8	4	5	0101
9	5	0	00000
10	5	1	00001

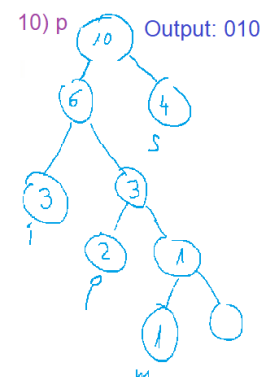
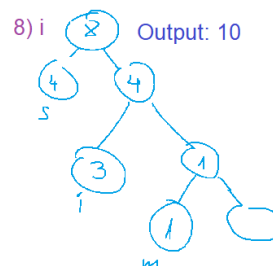
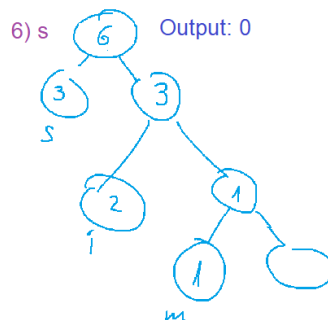
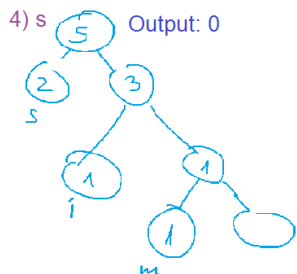
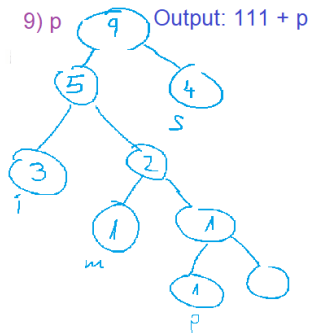
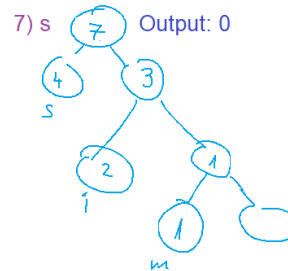
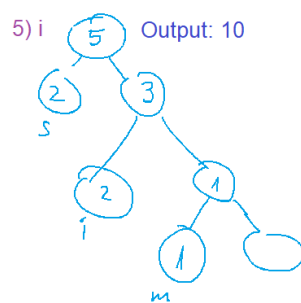
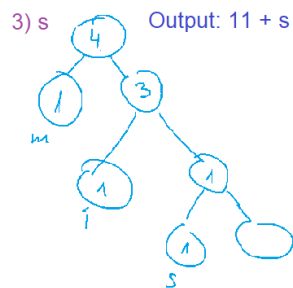
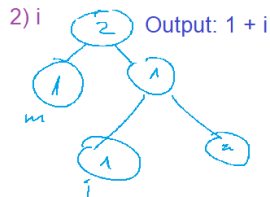
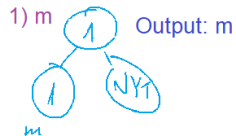
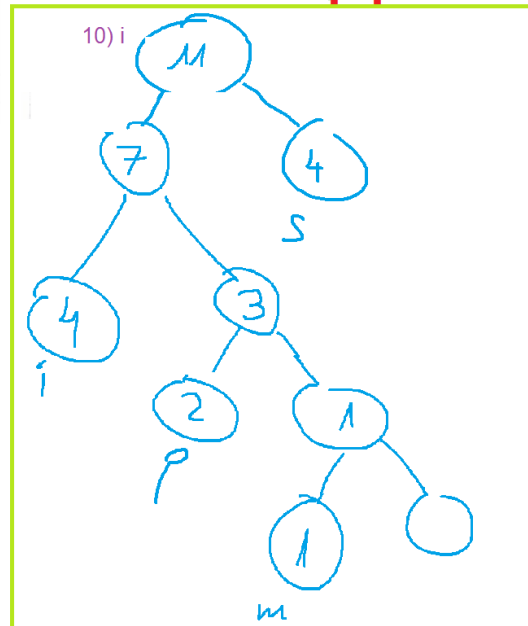
11	5	2	00010
12	5	3	00011

```

q = leaf(xi)
if (q is the 0-node)
    replace q by a parent 0-node with two 0-node children;
    q = left child;
if q is a sibling of a 0-node
    interchange q with the highest numbered leaf of the
    same weight;
    increment q's weight by 1;
    q = parent of q
while q ≠ root
    interchange q with the highest numbered node of the
    same weight;
    increment q's weight by 1;
    q = parent of q
increment a's weight by 1.

```

mississippi



## Arithmetic Code

- האנטרופיה של ההודעה בקידוד אריתמטי, היא כמות האינפורמציה של האינטרוול האחרון.

8 (א) עבור ההודעה:  $a^{999}b$  (מחרוזת בגודל 1000 – 999 a'ים ו-b אחד),  
מה גודל האינטרוול האחרון בקידוד אריתמטי?

גודל האינטרוול האחרון הוא מכפלת ההסתברויות של כל תו בקידוד ההודעה.  
גודל ההודעה הוא 1000,

$a$  מופיע 999 פעמים ולכן ההסתברות שלו היא  $\frac{999}{1000}$   
 $b$  מופיע פעם אחת ולכן ההסתברות שלו היא  $\frac{1}{1000}$

לכן מכפלת ההסתברויות הוא  $\left(\frac{999}{1000}\right)^{999} * \frac{1}{1000} = 3.680634883 * 10^{-4}$

(ב) מהו מספר הסיביות המינימאלי הנחוץ לקידוד הודעה זו בקוד אריתמטי?

מספר הסיביות המינימאלי הוא האנטרופיה.  
מכיוון שגודל האינטרוול האחרון הוא כבר מכפלת ההסתברויות, נפעיל עליו את האינפורמציה.

האנטרופיה היא  $-\log_2\left(\left(\frac{999}{1000}\right)^{999} * \frac{1}{1000}\right) = 11.40775774$

מכיוון שמדובר בביטים נעגל כלפי מעלה ונקבל ש-12 הוא מספר הסיביות המינימאלי הנחוץ לקידוד ההודעה.

(9 א) נתון הא"ב  $\Sigma = \{a, b\}$  בהסתברויות  $P = \{0.8, 0.2\}$  בהתאמה, ונתונה המחרוזת הבאה:  $aaaab$ .  
חשב מבחינה תיאורטית את מספר הסיביות המינימאלי הנחוץ לקידוד האינטרוול בקידוד אריתמטי.

גודל האינטרוול האחרון הוא מכפלת ההסתברויות.  
כמות האינפורמציה של מכפלת ההסתברויות הוא האנטרופיה (מספר הסיביות המינימאלי).

האנטרופיה היא  $-\log_2(0.8^4 * 0.2) = 3.609640474$ .  
מכיוון שמדובר בביטים נעגל כלפי מעלה ונקבל ש-4 הוא מספר הסיביות המינימאלי הנחוץ לקידוד ההודעה.

(ב) חשב ערך סופי היכול לשמש לקידוד ההודעה ע"פ קידוד אריתמטי.

M[i]	low	high	range
	0	1	1
a	$0 + 0 * 1 = 0$	$0 + 0.8 * 1 = 0.8$	0.8
a	$0 + 0 * 1 = 0$	$0 + 0.8 * 0.8 = 0.64$	0.64
a	$0 + 0 * 1 = 0$	$0 + 0.8 * 0.64 = 0.512$	0.512
a	$0 + 0 * 1 = 0$	$0 + 0.8 * 0.512 = 0.4096$	0.4096
b	$0 + 0.8 * 0.4096 = 0.32768$	$0 + 1 * 0.4096 = 0.4096$	0.08192

נצטרך להוציא פלט בין ה-Low וה-High האחרונים, לכן נוציא 0.35

(10) נתון הא"ב  $\{a, b, c, d, e, f\}$  בהסתברויות  $\{0.2, 0.3, 0.1, 0.2, 0.1, 0.1\}$  בהתאמה.

(א) בהנחה שההסתברויות בלתי תלויות, מהי ההסתברות למופע של המחרוזת  $baccf$  במחרוזת באורך 5?

פתרון:  $0.3 * 0.2 * 0.1 * 0.1 * 0.1 = 0.00006$

(ב) השתמש בקוד אריתמטי על מנת לקודד את המחזות  $.baccf$ .

פתרון:

$a - [0, 0.2), \quad b - [0.2, 0.5), \quad c - [0.5, 0.6), \quad d - [0.6, 0.8), \quad e - [0.8, 0.9), \quad f - [0.9, 1]$

נצטרך לתת מספר בין 0.23354 ל-0.2336.  
לכן, 0.23355

Char	Low	High	Range
	0	1	1
b	$0 + 0.2 * 1 = 0.2$	$0 + 0.5 * 1 = 0.5$	$0.5 - 0.2 = 0.3$
a	$0.2 + 0 * 0.3 = 0.2$	$0.2 + 0.2 * 0.3 = 0.26$	$0.26 - 0.2 = 0.06$
c	$0.2 + 0.5 * 0.06 = 0.23$	$0.2 + 0.6 * 0.06 = 0.236$	$0.236 - 0.23 = 0.006$
c	$0.23 + 0.5 * 0.006 = 0.233$	$0.23 + 0.6 * 0.006 = 0.2336$	$0.2336 - 0.233 = \frac{3}{5000}$
f	$0.233 + 0.9 * \frac{3}{5000} = 0.23354$	$0.233 + 1 * \frac{3}{5000} = 0.2336$	$0.2336 - 0.23354 = \frac{3}{50000}$

(ג) האם קיים קשר בין האינטרוול האחרון להסתברויות?

פתרון: כן, ה-Range של האינטרוול האחרון הוא מכפלת ההסתברויות.  
Adaptive Arithemtic Code

עבור א"ב עם 4 אותיות ההסתברות של האות  $x$  היא:

$$p(x) = \frac{N(x) + 1}{N(a) + N(b) + N(c) + N(d) + 4}, \quad x \in \Sigma$$

שאלות

(1) נתון ה-א"ב  $\{a, b, c, d\}$  וברצוננו לקודד את ההודעה bad באמצעות דחיסה אריתמטית אדפטיבית.

(א) השלם את הטבלה הבאה הנבנית במהלך אלגוריתם הדחיסה:

תו	$p(a)$	$p(b)$	$p(c)$	$p(d)$	low	high	Range
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	1	1
b	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$0 + \frac{1}{4} \cdot 1 = \frac{1}{4}$	$0 + \frac{1}{2} \cdot 1$	$\frac{1}{2} - \frac{1}{4} = \frac{1}{4}$
a	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{4} + \frac{1}{4} \cdot 0 = \frac{1}{4}$	$\frac{1}{4} + \frac{1}{4} \cdot \frac{1}{5} = \frac{3}{10}$	$\frac{3}{10} - \frac{1}{4} = \frac{1}{20}$
d	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{4} + \frac{1}{20} \cdot \frac{5}{6} = \frac{7}{24}$	$\frac{1}{4} + \frac{1}{20} \cdot 1 = \frac{3}{10}$	$\frac{3}{10} - \frac{7}{24} = \frac{1}{120}$

(ב) רשום את הטווח האחרון כמכפלת ההסתברויות המתאימה.

פתרון:  $\frac{1}{4} \cdot \frac{1}{5} \cdot \frac{1}{6} = \frac{1}{120}$

(2) נתון הא"ב הסדור  $\Sigma = \{a, z, o\}$  וברצוננו לפענח את ההודעה 0.64 באמצעות פענוח של דחיסה אריתמטית אדפטיבית. הנח/י התפלגות אחידה של תווי הא"ב.

א. בנוסף לא"ב הסדור וההודעה הדחוסה, על המקודד לשלוח מידע נוסף. מהו אותו מידע?

ב. פענח את ההודעה. הנח כי המידע החסר הוא 4.

**תזכורת:** עבור א"ב עם 3 אותיות ההסתברות של האות  $x$  היא :

$$p(x) = \frac{N(x)+1}{N(a)+N(z)+N(o)+3}, x \in \Sigma$$

כאשר  $N(x)$  הוא מספר המופעים של  $x$  בקטע המחרוזת שהתקבל.

פתרון:

$$E = 0.64$$

Char	$P(a)$	$P(z)$	$P(o)$	Low	High	Range
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	1	1
z	$\frac{0+1}{0+1+0+3} = \frac{1}{4}$	$\frac{1+1}{0+1+0+3} = \frac{2}{4}$	$\frac{0+1}{0+1+0+3} = \frac{1}{4}$	$0 + \frac{1}{3} * 1 = \frac{1}{3}$	$0 + \frac{2}{3} * 1 = \frac{2}{3}$	$\frac{1}{3}$
o	$\frac{0+1}{0+1+1+3} = \frac{1}{5}$	$\frac{1+1}{0+1+1+3} = \frac{2}{5}$	$\frac{1+1}{0+1+1+3} = \frac{1}{5}$	$\frac{1}{3} + \frac{1}{4} * \frac{1}{3} = \frac{7}{12}$	$\frac{1}{3} + 1 * \frac{1}{3} = \frac{2}{3}$	$\frac{1}{12}$

$o$	$\frac{0+1}{0+1+2+3} = \frac{1}{6}$	$\frac{1+1}{0+1+2+3} = \frac{2}{6}$	$\frac{2+1}{0+1+2+3} = \frac{3}{6}$	$\frac{7}{12} + \frac{3}{5} * \frac{1}{12} = \frac{19}{30}$	$\frac{7}{12} + 1 * \frac{1}{12} = \frac{2}{3}$	$\frac{1}{30}$
$z$	$\frac{0+1}{0+2+2+3} = \frac{1}{7}$	$\frac{2+1}{0+2+2+3} = \frac{3}{7}$	$\frac{2+1}{0+2+2+3} = \frac{3}{7}$	$\frac{13}{20}$	$\frac{23}{36}$	

1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{13}{20}$
$o$	$o$	$o$	$o$	
$\frac{2}{3}$	$\frac{1}{3} + \frac{3}{4} * \frac{1}{3} = \frac{7}{12}$	$\frac{7}{12} + \frac{3}{5} * \frac{1}{12} = \frac{19}{30}$	$\frac{19}{30} + \frac{3}{6} * \frac{1}{30} = \frac{13}{20}$	
$z$	$z$	$z$	$z$	
$\frac{1}{3}$	$\frac{1}{3} + \frac{1}{4} * \frac{1}{3} = \frac{5}{12}$	$\frac{7}{12} + \frac{1}{5} * \frac{1}{12} = \frac{3}{5}$	$\frac{19}{30} + \frac{1}{6} * \frac{1}{30} = \frac{23}{36}$	
$a$	$a$	$a$	$a$	
0	$\frac{1}{3}$	$\frac{7}{12}$	$\frac{19}{30}$	$\frac{23}{36}$

$o \left[ \frac{2}{3}, 1 \right)$	$o \left[ \frac{3}{4}, 1 \right)$	$o \left[ \frac{3}{5}, 1 \right)$	$o \left[ \frac{3}{6}, 1 \right)$
$z \left[ \frac{1}{3}, \frac{2}{3} \right)$	$z \left[ \frac{1}{4}, \frac{3}{4} \right)$	$z \left[ \frac{1}{5}, \frac{3}{5} \right)$	$z \left[ \frac{1}{6}, \frac{3}{6} \right)$
$a \left[ 0, \frac{1}{3} \right)$	$a \left[ 0, \frac{1}{4} \right)$	$a \left[ 0, \frac{1}{5} \right)$	$a \left[ 0, \frac{1}{6} \right)$

Re-Pair

אלגוריתם קידוד :

- לולאה – כל עוד אין זוג נוסף בטקסט.
  - מצא את הזוג  $ab$  המופיע הכי הרבה פעמים בטקסט.
  - צור עבור הזוג חוק גזירה  $A \rightarrow ab$ .
  - החלף את כל הזוגות המופיעים בטקסט בחוק הגזירה  $A$ .

Sequiter



חוקים :

- $T = \text{singing\_do\_wah\_diddy\_diddy\_dum\_diddy\_do}$ 
  - (1) אין זוג של תווים שחוזר על עצמו.
  - (2) כל חוק גזירה צריך להיות בשימוש יותר מפעם אחת.

$A \rightarrow \_d$	singingAo_wahAidddyAidddyAumAidddyAo	דוגמא :
$B \rightarrow dd$	singingAo_wahAiByAiByAumAiByAo	
$C \rightarrow Ai$	singingAo_wahCByCByAumCByAo	
$D \rightarrow Dum$	singingAo_wahCByCByAumCByAo	

נרצה לקודד את המחזורות – "abcbcbabcbcb".

<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow abcbdb</math></li></ul></li></ul>		
<ul style="list-style-type: none"><li>abcbcbabcbcb -&gt; Rule 1<ul style="list-style-type: none"><li><math>S \rightarrow abcbdb</math></li></ul></li></ul>	<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow aAdA</math></li><li><math>A \rightarrow bc</math></li></ul></li></ul>	
<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow aAdAa</math></li><li><math>A \rightarrow bc</math></li></ul></li></ul>		
<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow aAdAab</math></li><li><math>A \rightarrow bc</math></li></ul></li></ul>		
<ul style="list-style-type: none"><li>abcbcbabcbcb -&gt; Rule 1<ul style="list-style-type: none"><li><math>S \rightarrow aAdAaA</math></li><li><math>A \rightarrow bc</math></li></ul></li></ul>	<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow BdAB</math></li><li><math>A \rightarrow bc</math></li><li><math>B \rightarrow aA</math></li></ul></li></ul>	
<ul style="list-style-type: none"><li>abcbcbabcbcb -&gt; Rule 1<ul style="list-style-type: none"><li><math>S \rightarrow BdABd</math></li><li><math>A \rightarrow bc</math></li><li><math>B \rightarrow aA</math></li></ul></li></ul>	<ul style="list-style-type: none"><li>abcbcbabcbcb -&gt; Rule 2<ul style="list-style-type: none"><li><math>S \rightarrow CAC</math></li><li><math>A \rightarrow bc</math></li><li><math>B \rightarrow aA</math></li><li><math>C \rightarrow Bd</math></li></ul></li></ul>	<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow CAC</math></li><li><math>A \rightarrow bc</math></li><li><math>C \rightarrow aAd</math></li></ul></li></ul>
<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow CACb</math></li><li><math>A \rightarrow bc</math></li><li><math>C \rightarrow aAd</math></li></ul></li></ul>		
<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow CACbc</math></li><li><math>A \rightarrow bc</math></li><li><math>C \rightarrow aAd</math></li></ul></li></ul>	<ul style="list-style-type: none"><li>abcbcbabcbcb<ul style="list-style-type: none"><li><math>S \rightarrow CACA</math></li><li><math>A \rightarrow bc</math></li><li><math>C \rightarrow aAd</math></li></ul></li></ul>	

LZSS

שאלות

(1) הצג את הקובץ הדחוס שנוצר באמצעות אלגוריתם LZSS עבור ההודעה \_wed\_we\_wee\_web\_wet מה גודלו (במספר סיביות) של הקובץ כולו, בהנחה שרכיב הצבעה מכיל 12 ביטים ורכיב תו בודד מכיל 8 סיביות?

הקובץ הדחוס: \_,w,e,d,(4,3),(3,3),(4,4),b,(4,3),t  
מספר הסיביות:  $8*6+12*4=96$



LZW

LZW Terry Encoding Algorithm

```
1. Dictionary ← single Characters
2. w ← first char of input
3. repeat{
  1. k ← next char
  2. if (EOF)
    1. output code(w)
  3. else if (w · k) ∈ Dictionary
    1. w ← w · k
  4. else
    1. output code(w)
    2. Dictionary ← w · k
    3. w ← k
}
```

LZW Decoding Algorithm

```
1. Initialize table with single character strings
2. OLD = first input code
3. output translation of OLD
4. while not end of input stream{
  1. NEW = next input code
  2. if NEW is not in the string table
    1. S = translation of OLD
    2. S = S · C
  3. else
    1. S = translation of NEW
  4. output S
  5. C = first character of S
  6. Translation(OLD) · C to the string table
  7. OLD = NEW
}
```

<ul style="list-style-type: none"><li>• אם W+K במילון<ul style="list-style-type: none"><li>○ W יהיה W+K</li></ul></li><li>• אם W+K לא במילון<ul style="list-style-type: none"><li>○ נפלוט את הקוד של W</li><li>○ נוסיף למילון את W+K</li><li>○ W יהיה שווה ל-K</li></ul></li><li>• אם סיימנו את הקובץ – נפלוט את הקוד האחרון.</li></ul>	<ul style="list-style-type: none"><li>• אם NEW לא במילון<ul style="list-style-type: none"><li>○ S יהיה התרגום של OLD+C</li></ul></li><li>• אם NEW במילון<ul style="list-style-type: none"><li>○ S יהיה התרגום של NEW</li></ul></li><li>• C יהיה התו הראשון של S</li><li>• נוסיף למילון את התרגום של OLD+C</li></ul>
---	---

שאלות

מספר כניסה	מחרוזת
0	"A"
1	"B"
2	"C"
...	...

1) נניח כי המילון בו משתמשים באלגוריתם LZW מאותחל ע"פ הטבלה הבאה:

א) פרש את ההודעה הבאה (משמאל לימין):  
0,1,3,5,6,2,7,9,2

הראה את המילון בכל שלב ושלב של האלגוריתם.

					D.EntryNum	Dictionary	עבור סעיף ב'
					0	A	00
i	OLD	NEW	S	C	1	B	01
	0		A		2	C	10
1	0	1	B	B	3	AB	11
2	1	3	AB	A	4	BA	100
3	3	5	ABA	A	5	ABA	101
4	5	6	ABAA	A	6	ABAA	110
5	6	2	C	C	7	ABAAC	111
6	2	7	ABAAC	A	8	CA	1000
7	7	9	ABAACA	A	9	ABAACA	1001
8	9	2	C	C	10	ABAACAC	1010

ב) בהנחה כי מספר הכניסות הראשוני היה 4 ומוכפל בכל פעם שהוא מתמלא, מה אורכה של ההודעה מבחינת הסיביות?

- היו לנו 3 מילים בהתחלה עבור 3 כניסות (0-2).  
באיטרציה 1, מילאנו את כניסה 3 (הרביעית), לכן המילון הוכפל.  
עד כאן יש לנו 4 כניסות שאת אותן ספרות 0-3 ניתן לקודד באמצעות 2 סיביות.  
4 כניסות \* 2 סיביות.
- באיטרציה 5 קראנו את 2 ומילאנו את הכניסה 7 (השמינית), לכן המילון הוכפל.  
כעת נצטרך לקודד את מספרי הכניסות 4-7 אותן ניתן לקודד עם 3 סיביות.  
4 כניסות \* 3 סיביות.  
באיטרציה 8 סיימנו לאחר שקראנו את 2 ומילאנו את כניסה 10.  
3 כניסות \* 4 סיביות.

לכן,  $32 = 2 * 4 + 3 * 4 + 4 * 3$  כניסות.

(2) נתונה המחזורת הבאה: "\_wed\_we\_wee\_web\_wet"  
הצג את המילון המתקבל מהפעלת דחיסת LZW ואת הקובץ הדחוס. הנח כי המילון  
ההתחלתי מכיל את כל 256 אותיות ה-ASCII. כמה סיביות יכול קובץ זה בהנחה שהמילון  
אלגוריתם יכול להכיל 512 כניסות?

**פתרון:**

- $n =$  אורך הביטים לקידוד
- $D =$  המילון של  $w$  - נסמן ב- $w$  את המספר ב-ASCII של  $w$ , ב- $w'$  את ה-ASCII של  $w$
- $\Sigma =$  האותיות של  $w$  - המילון לכל תו את ה-ASCII שלו.

**Algorithm**

```
Tunstall(n) {  
  D ← Σ  
  while (|D| ≤ 2^n - |Σ|) {  
    Let d ∈ D be with highest probability  
    D ← D ∪ {dσ} (σ ∈ Σ)  
    if d ∉ Σ  
      D ← D - {d}  
  }  
}
```

k	Output	Dictionary
w	'_'	נכנס למילון את 256
e		לולאה - כל המילון קטן מ $2^n -  Σ $
d		נכנס $d$ עם ההסתברות הגבוהה ביותר.
-		נכנס למילון את השייך של $d$ עם כל ה-א"ב.
w		ההסתברות של השיר שורים היא מכפלה ההסתברויות.
e		אם $d$ לא שייך ל-א"ב, נוציא את $d$ מהמילון.
e		256
-		260 = _we
w		261 = e_
e		262 = _w
-		263 = e_w
w		264 = web
e		265 = _w
-		266 = _we
w		267 = wet
e		268 = _w
-		269 = e_w
w		270 = _w
e		271 = e_w
-		272 = _w
w		273 = e_w
e		274 = _w
-		275 = e_w
w		276 = _w
e		277 = e_w
-		278 = _w
w		279 = e_w
e		280 = _w
-		281 = e_w
w		282 = _w
e		283 = e_w
-		284 = _w
w		285 = e_w
e		286 = _w
-		287 = e_w
w		288 = _w
e		289 = e_w
-		290 = _w
w		291 = e_w
e		292 = _w
-		293 = e_w
w		294 = _w
e		295 = e_w
-		296 = _w
w		297 = e_w
e		298 = _w
-		299 = e_w
w		300 = _w
e		301 = e_w
-		302 = _w
w		303 = e_w
e		304 = _w
-		305 = e_w
w		306 = _w
e		307 = e_w
-		308 = _w
w		309 = e_w
e		310 = _w
-		311 = e_w
w		312 = _w
e		313 = e_w
-		314 = _w
w		315 = e_w
e		316 = _w
-		317 = e_w
w		318 = _w
e		319 = e_w
-		320 = _w
w		321 = e_w
e		322 = _w
-		323 = e_w
w		324 = _w
e		325 = e_w
-		326 = _w
w		327 = e_w
e		328 = _w
-		329 = e_w
w		330 = _w
e		331 = e_w
-		332 = _w
w		333 = e_w
e		334 = _w
-		335 = e_w
w		336 = _w
e		337 = e_w
-		338 = _w
w		339 = e_w
e		340 = _w
-		341 = e_w
w		342 = _w
e		343 = e_w
-		344 = _w
w		345 = e_w
e		346 = _w
-		347 = e_w
w		348 = _w
e		349 = e_w
-		350 = _w
w		351 = e_w
e		352 = _w
-		353 = e_w
w		354 = _w
e		355 = e_w
-		356 = _w
w		357 = e_w
e		358 = _w
-		359 = e_w
w		360 = _w
e		361 = e_w
-		362 = _w
w		363 = e_w
e		364 = _w
-		365 = e_w
w		366 = _w
e		367 = e_w
-		368 = _w
w		369 = e_w
e		370 = _w
-		371 = e_w
w		372 = _w
e		373 = e_w
-		374 = _w
w		375 = e_w
e		376 = _w
-		377 = e_w
w		378 = _w
e		379 = e_w
-		380 = _w
w		381 = e_w
e		382 = _w
-		383 = e_w
w		384 = _w
e		385 = e_w
-		386 = _w
w		387 = e_w
e		388 = _w
-		389 = e_w
w		390 = _w
e		391 = e_w
-		392 = _w
w		393 = e_w
e		394 = _w
-		395 = e_w
w		396 = _w
e		397 = e_w
-		398 = _w
w		399 = e_w
e		400 = _w
-		401 = e_w
w		402 = _w
e		403 = e_w
-		404 = _w
w		405 = e_w
e		406 = _w
-		407 = e_w
w		408 = _w
e		409 = e_w
-		410 = _w
w		411 = e_w
e		412 = _w
-		413 = e_w
w		414 = _w
e		415 = e_w
-		416 = _w
w		417 = e_w
e		418 = _w
-		419 = e_w
w		420 = _w
e		421 = e_w
-		422 = _w
w		423 = e_w
e		424 = _w
-		425 = e_w
w		426 = _w
e		427 = e_w
-		428 = _w
w		429 = e_w
e		430 = _w
-		431 = e_w
w		432 = _w
e		433 = e_w
-		434 = _w
w		435 = e_w
e		436 = _w
-		437 = e_w
w		438 = _w
e		439 = e_w
-		440 = _w
w		441 = e_w
e		442 = _w
-		443 = e_w
w		444 = _w
e		445 = e_w
-		446 = _w
w		447 = e_w
e		448 = _w
-		449 = e_w
w		450 = _w
e		451 = e_w
-		452 = _w
w		453 = e_w
e		454 = _w
-		455 = e_w
w		456 = _w
e		457 = e_w
-		458 = _w
w		459 = e_w
e		460 = _w
-		461 = e_w
w		462 = _w
e		463 = e_w
-		464 = _w
w		465 = e_w
e		466 = _w
-		467 = e_w
w		468 = _w
e		469 = e_w
-		470 = _w
w		471 = e_w
e		472 = _w
-		473 = e_w
w		474 = _w
e		475 = e_w
-		476 = _w
w		477 = e_w
e		478 = _w
-		479 = e_w
w		480 = _w
e		481 = e_w
-		482 = _w
w		483 = e_w
e		484 = _w
-		485 = e_w
w		486 = _w
e		487 = e_w
-		488 = _w
w		489 = e_w
e		490 = _w
-		491 = e_w
w		492 = _w
e		493 = e_w
-		494 = _w
w		495 = e_w
e		496 = _w
-		497 = e_w
w		498 = _w
e		499 = e_w
-		500 = _w
w		501 = e_w
e		502 = _w
-		503 = e_w
w		504 = _w
e		505 = e_w
-		506 = _w
w		507 = e_w
e		508 = _w
-		509 = e_w
w		510 = _w
e		511 = e_w
-		512 = _w

דוגמא עבור  $n = 4$ :  $\Sigma = \{a, b, c, d\}$ ,  $P = \{0.5, 0.3, 0.1, 0.1\}$

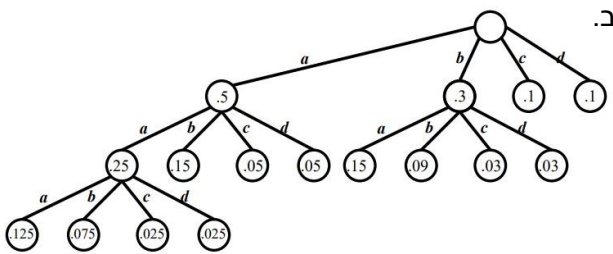
שלב 1: נכנסת  $a$  למילון. כרגע יש 4 מילים במילון.

שלב 2:  $4 \leq 12$ , נכנסים ללולאה.  $d = a$  (ההסתברות הגבוהה ביותר), לכן נשרשר אליו את ה-א"ב. במילון יש 7 מילים.

שלב 3:  $7 \leq 12$ , נכנסים ללולאה.  $d = b$  (ההסתברות הגבוהה ביותר), לכן נשרשר אליו את ה-א"ב. במילון יש 10 מילים.

שלב 4:  $10 \leq 12$ , נכנסים ללולאה.  $d = aa$  (ההסתברות הגבוהה ביותר), לכן נשרשר אליו את ה-א"ב. במילון יש 13 מילים.

שלב 5:  $13 > 12$ , לכן נצא מהלולאה.



Dictionary word	Codeword	Dictionary word	Codeword
aaa	0000	bb	1000
aab	0001	bc	1001
aac	0010	bd	1010
aad	0011	a	1011
ab	0100	b	1100
ac	0101	c	1101
ad	0110	d	1110
ba	0111		

Fibonacci

ייצוג בינארי של פיבונאצ'י:

- נדליק בכל פעם את הביט הגדול ביותר שנכנס למספר.

Fibonacci Binary Representation - Example #1

Basis elements:

128 64 32 16 8 4 2 1

73 = 0 1 0 0 1 0 0 1

Fibonacci:

55 34 21 13 8 5 3 2 1

73 = 1 0 0 1 0 1 0 0 0

Fibonacci Binary Representation - Example #2

Basis elements:

128 64 32 16 8 4 2 1

45 = 0 0 1 0 1 1 0 1

Fibonacci:

55 34 21 13 8 5 3 2 1

45 = 0 1 0 0 1 0 1 0 0

Fib1:

- ❖ Example: 19 = 1<sup>13</sup>101001
- ❖ Problem: Not instantaneous

1 1 0 1 1 1 1 1 1 1 0

- ❖ Solution: Reverse the codeword
- ❖ Example: 19 = 1001011

- נייצג את המספר בייצוג בינארי של פיבונאצ'י.
- נוסיף 1 מוביל.
- נהפוך את הקוד.

Fib2:

- נקודד באמצעות Fib1.
- נוריד את ה-1 שהוספנו (לאחר הרוורס הוא בצד ימין).
- נוסיף 10 בהתחלה.
- \*הערה - ב-Fib2 נייצג את הספרה 1 על ידי הקוד '1'.

i	Fibonacci Binary Representation	Fib1	Fib2