דחיסת נתונים – סיכום

הגדרה (קוד חסר-ראשות). אף מילה אינה רישא של מילת קוד אחרת.
הגדרה (קוד ניתן-לפיענוח). כל סדרה של מילות-קוד ניתן לקודד לפענח באופן אחד.
הגדרה (קוד שלם). כל סדרה אינסופית למחצה ניתנת לפיענוח בצורה יחידה. **קוד שלם** $\Longleftrightarrow$ **עץ מלא**.
הגדרה (קוד מיידי)-בסוף כל מילת קוד יודעים (המפענח יודע) שזה סופה. **מיידי** $\Longleftrightarrow$ **חסר ראשות.**

משפט (אי שוויון קראפט)

עבור קוד $C=\{c_1,\dots,c_n\}$ נגדיר $K(C)=\sum_{i=1}^{n}2^{-|c_i|}$ אזי מתקיים
1. אם $K(C)\le 1$ אזי C ניתן לפיענוח.
2. אם $K(C)\le 1$ אזי קיים קוד C' חסר רישות כך ש-$\mathbb{E}(C,P)=\mathbb{E}\left(C',P\right)$.

מודלים

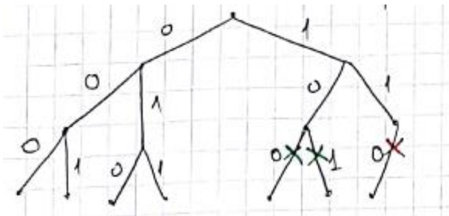
ישנם מספר סוגים של מודלים:

מודל מרקוב מסדר ראשון

ישנם מספר סוגי מודלים הנופלים תחת קטגוריה זאת:

- **קודים סטטיים** – שיטות קידוד סטטיות הם פשטניות למדי, הקידודים הללו ידועים לכולם (כגון קידוד הסקי). במודל זה יש שימוש מינימלי בשכיחויות של התווים, השכיחויות נתונות רק לצורך מיון בסדרה מונוטוניות יורדת. יעילות הדחיסה אינה מרשימה במיוחד אולם היות והיא פשוטה אז היא מאוד שימושית והיתרון הוא מהירות הפיענוח.
  - **קידוד אונארי** – מילת הקוד $1^{i-1}0$ מייצגת את מילת הקוד ה-$i\in\mathbb{N}$, כלומר את c_i.
  $$c_1=0, c_2=10, c_3=110, c_4=1110,\dots$$
  אם מספר המילים שאנחנו מקודדים הוא סופי $n\in\mathbb{N}$, ואנחנו יודעים מראש את כל המילים אותם אנו מקודדים נוכל לקחת את המילה שמופיעה הכי מעט כלומר c_n ובמקום לקודד אותה ב-$1^{n-1}0$ נקודד אותה ב-1^{n-1}, זאת אומרת שחסנו מילת תו אחד בכל קידוד של המילה. מצד שני, זה אינו באמת משמעותי שכן זוהי המילה הכי פחות שכיחה.
  נשים לב: קוד אונארי הוא בעל יתירות מינימלית אם ההסתברויות הם חזקות של 2, כלומר $p_i=1/2^{\ell_i}$, כאשר ℓ_i היא אורך מילת הקוד ה-i.
  $$\mathbb{E}(C)=\sum_{i=1}^{n}p_i\cdot\ell_i=-\sum_{i=1}^{n}p_i\cdot\log_2 p_i=H(C)\Longleftrightarrow\sum_{i=1}^{n}\ell_i=\sum_{i=1}^{n}-\log_2 p_i$$
  ע"פ ההנחה ש-$p_i=2^{-\ell_i}$ נובע כי $-\log_2 p_i=-\log_2 2^{-\ell_i}=\ell_i$, ולכן צד ימין מתקיים, לפיכך התוחלת מינימלית שכן היא האנטרופיה והאנטרופיה הינה חסם תחתון על אורך מילת הקוד הממוצעת. **הערה**. כל סידור של ההסתברויות הנ"ל יגדיר קוד בעל יתירות מינימלית.
  - **קוד דיאדי** – קוד שבו ההסתברויות הם חזקות של 2 וכמות האינפורמציה שווה בדיוק לגודל מילת-הקוד. בעצם ראינו שקוד אונארי הוא אופטימלי כאשר הוא קוד דיאדי.

- **קוד בינארי פשוט –** קוד בעל אורך קבוע כאשר מילת הקוד c_i מקודדת ע"פ הקידוד הבינארי של $i \in \mathbb{N}$. אם יש לנו n מילות קוד, אזי מספר התווים הנדרשים יהיו $\lceil \log_2 n \rceil$. הבעייתיות בקוד הנ"ל היא שאם יש לנו 7 מילות קוד נקבל את העץ הבא:



ביצוע קיצוץ לעץ בצורה המתוארת לעיל מעלה את ההצעה הבאה:

- **קוד בינארי מינימלי -** עבור קידוד של n מילים נבחר את $n - 2^{\lfloor \log_2 n \rfloor}$ מילות הקוד הראשונות (בעלי ההסתברויות הגבוהות ביותר) להיות באורכים $\lceil \log_2 n \rceil$, ואת שאר המילים להיות באורך $\lfloor \log_2 n \rfloor$.

הערה מאיפה הגיעו המספרים האלה. יש $n - 2^{\lfloor \log_2 n \rfloor}$ מילים מיותרות בעץ שלם בנוסף כיוון שבעץ מינימלי כל מילה באורך $\lceil \log_2 n \rceil$ תופסת שתי מקומות מתבקש להסיק כי כל מילה כזאת תתפוס את המקום שלה ועוד מקום של מילה מיותרת, היות ויש $n - 2^{\lfloor \log_2 n \rfloor}$ מילים מיותרת נוכל להקצות לכל היותר $n - 2^{\lfloor \log_2 n \rfloor}$ מילים באורך $\lceil \log_2 n \rceil$. וכן סך המילים:

$$2^{\lceil \log_2 n \rceil} = 2\left(2^{\lceil \log_2 n \rceil} - n\right) - \left(n - \left(2^{\lceil \log_2 n \rceil} - n\right)\right) = 2 \cdot 2^{\lceil \log_2 n \rceil} - 2n + 2n - 2^{\lceil \log_2 n \rceil}$$

כנדרש.

- **קוד בינארי מינימלי הוא אופטימלי כאשר** $p_i = 2^{-\ell_i}$.

- **קידוד C_γ –** הקידוד מורכב משני קידוד, החלק הראשון הוא קידוד אונארי של אורך הקידוד הבינארי של $i \in \mathbb{N}$, כלומר $\lfloor \log_2 i \rfloor + 1$ באונארית. החלק השני הוא הקידוד הבינארי של i כאשר משמיטים את ה-MSB (most significant bit) מהקידוד הבינארי. ההשמטה של ה-MSB מתבצעת כיוון שהוא תמיד 1 ולכן מיותר (הוכח באינדוקציה). נשים לב שמספר הביטים שנדרשו הינו $1 + 2 \cdot \lfloor \log_2 n \rfloor$. נשים לב שבשביל למצוא מילה בקידוד צריך ללכת לקידוד האונרי של אורך הקידוד הבינארי ואז לבצע חיפוש בינארי בתוך אותו חלק. חלק ראשון $Unary(|Binary(i)|)$, חלק שני $Binary^*(i)$.

x	Elias C_γ Code
1	0
2	10 0
3	10 1
4	110 00
5	110 01
6	110 10
7	110 11
8	1110 000
9	1110 001

Also called:
"exponential and binary search" – Bentley and Yao (1976)

- **קידוד C_δ –** שני חלקים: קידוד של הערך בבינארית של האינדקס $i \in \mathbb{N}$ באמצעות C_γ כלומר $C_\gamma(|Binary(i)|)$, והחלק השני הוא i בבינארית בלי ה-MSB, כלומר $Binary^*(i)$. על מנת לספור את מספר הביטים שנדרשים כדי לקודד את c_i, נניח ש-y הוא הקידוד הבינארי של i, ולכן לקודד את y ב-C_γ לוקח $1 + 2\lfloor 1 + 2\log_2 i \rfloor = 1 +$ $2\lfloor \log_2 \log_2 2i \rfloor$והחלק השני לוקח $\lfloor \log_2 i \rfloor$. סה"כ

$$\|\langle c_i \rangle_\delta\| = 1 + \lfloor \log_2 i \rfloor + 2\lfloor \log_2 \log_2 2i \rfloor$$

x	Elias C_δ Code
1	0
2	100 0
3	100 1
4	101 00
5	101 01
6	101 10
7	101 11
8	11000 000
9	11000 001

- הבעיה העיקרית בשני הקודים הללו C_γ, C_δ היא שהדליים המתאימים לחלק ספציפי בקידוד האונארי גדלים אקספוננציאלית ולכן הציעו את קידוד *Golomb* עבור דלי בגודל קבוע b.

- **קידוד *Golomb* –** לשם שינוי נתחיל מדוגמה נניח אנחנו רוצים לקודד את c_8 וגודל הדלי הוא 5, נוכל להסיק ששמונה בדלי השני במיקום 3 בדלי, לכן הקידוד של הדלי ה-2 באונארית הינו 10 והקידוד של מילת הקוד השלישית בדלי שגודלו 5 הוא יהיה

```
Golomb_encode(x,b){
    q←(x-1) div b;
    r←x-q·b;
    Unary_encode(q+1);
    Minimal_binary_encode(r,b);
```

$MBE(3,5)$, כלומר מילת הקוד השלישית בקידוד בינארי מינימלי בעל 5 מילות קוד. היות ומילת הקוד
החמישית היא בדלי הראשון צריך מינוס אחד ב-q, וכיו"ב.

- **פיענוח גולומב –** מקבלים מילה לפיענוח α ואת גודל הדלי b, נשים לב שהאינדקס הוא
$$MBD(\alpha, b) + (UnaryD(\alpha) - 1) \cdot b$$
- **קוד רייס –** מקרה פרטי של גולמב כאשר הדליים הם חזקות של 2, ואז העץ בינארי המינימלי הוא גם העץ בינארי הפשוט וזה יקל עלינו קצת. לדוגמה ניקח $b = 2^2$ ואת האינדקס 8, אז המילה בדלי השני, במיקום 4 כלומר הקידוד 1011. מסיבות כאלו ואחרות יותר יעיל לבצע את החישוב באופן אחר שמתואר במחברת של תום. במקרה לעיל מתקבל
$$8 \to \langle 8 - 1 \rangle_2 \to 111 \to 1 \to \langle 1 \rangle_{10} + 1 = 2 \to \langle 2 \rangle_1 = 10 \to 1011$$
כאשר בסוף מוסיפים את האיברים שמחקנו, דוגמה נוספת:
$$i = 30, b = 2^3 \to \langle 30 - 1 \rangle_2 \to 11101 \to 11 \to \langle 11 \rangle_{10} + 1 = 3 + 1 = 4 \to \langle 4 \rangle_1 = 1110 \to 1110101$$

הגדרה (קוד בעל יתירות מינימלית)
נאמר שקוד C הוא בעל יתירות מינימלית אם לכל קוד אחר C' מתקיים $\mathbb{E}(C) \leq \mathbb{E}(C')$.

- **קוד שנון –** הקוד מקיים $\ell_i = -\lceil \log_2 p_i \rceil$, וניתן להוכיח שאורך מילת הקוד הממוצעת היא לכל היותר אחד ועוד האנטרופיה. מצד אחד זה די טוב, מצד שני במקרה הגרוע כל מילת קוד ארוכה בסיבית מהחסם התחתון.

למטה נסמן ב-$Binary^*$ את הפונקציה שמחזירה את הקידוד הבינארי ללא ה-MSB. וב-$\langle n \rangle_2$ את הקידוד הבינארי של n.

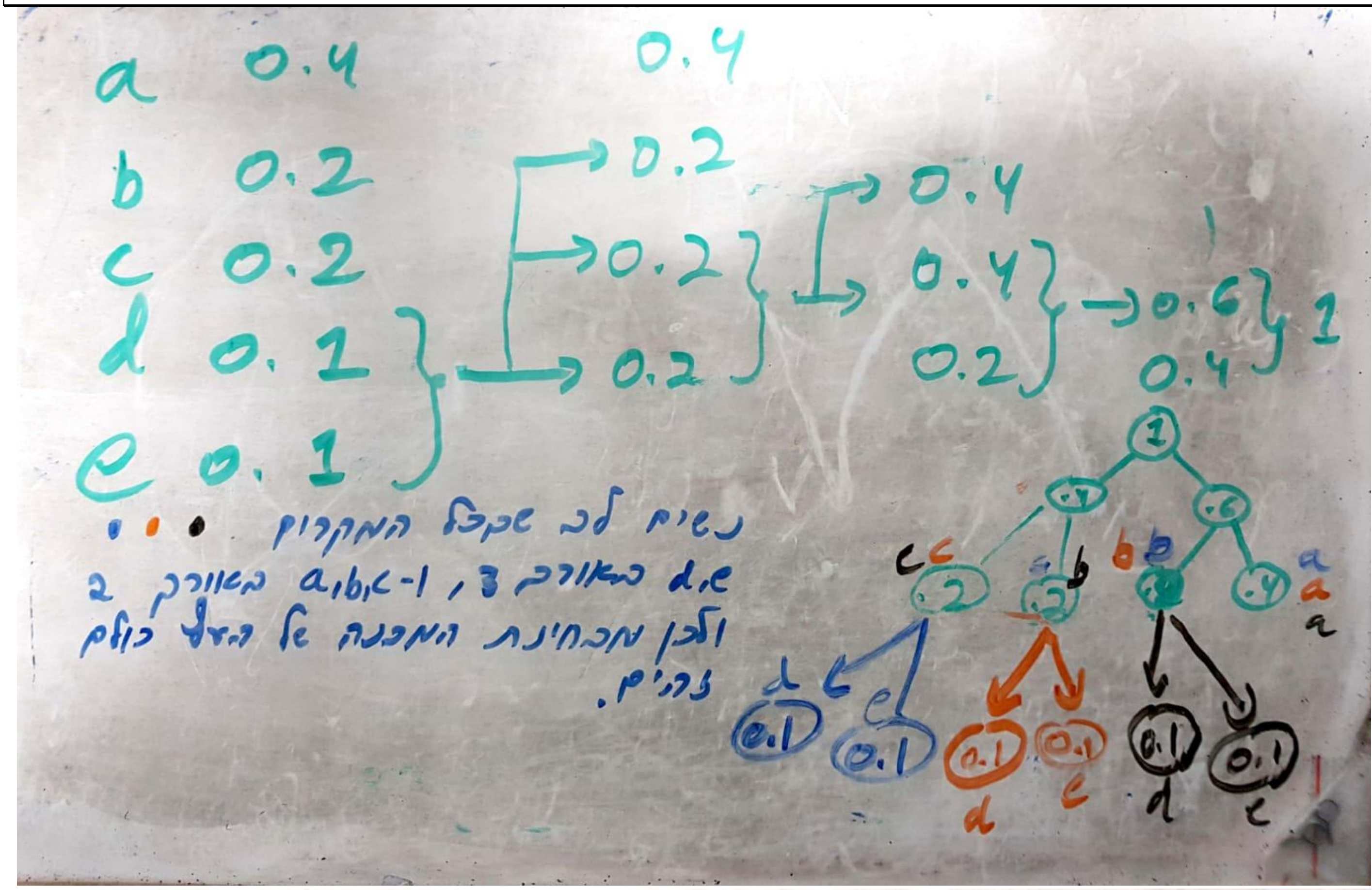
אלגוריתם לקידוד מילה (מספר) σ באמצעות C_γ: $Unary(|\langle \sigma \rangle_2|) \circ Binary^*(\sigma)$.

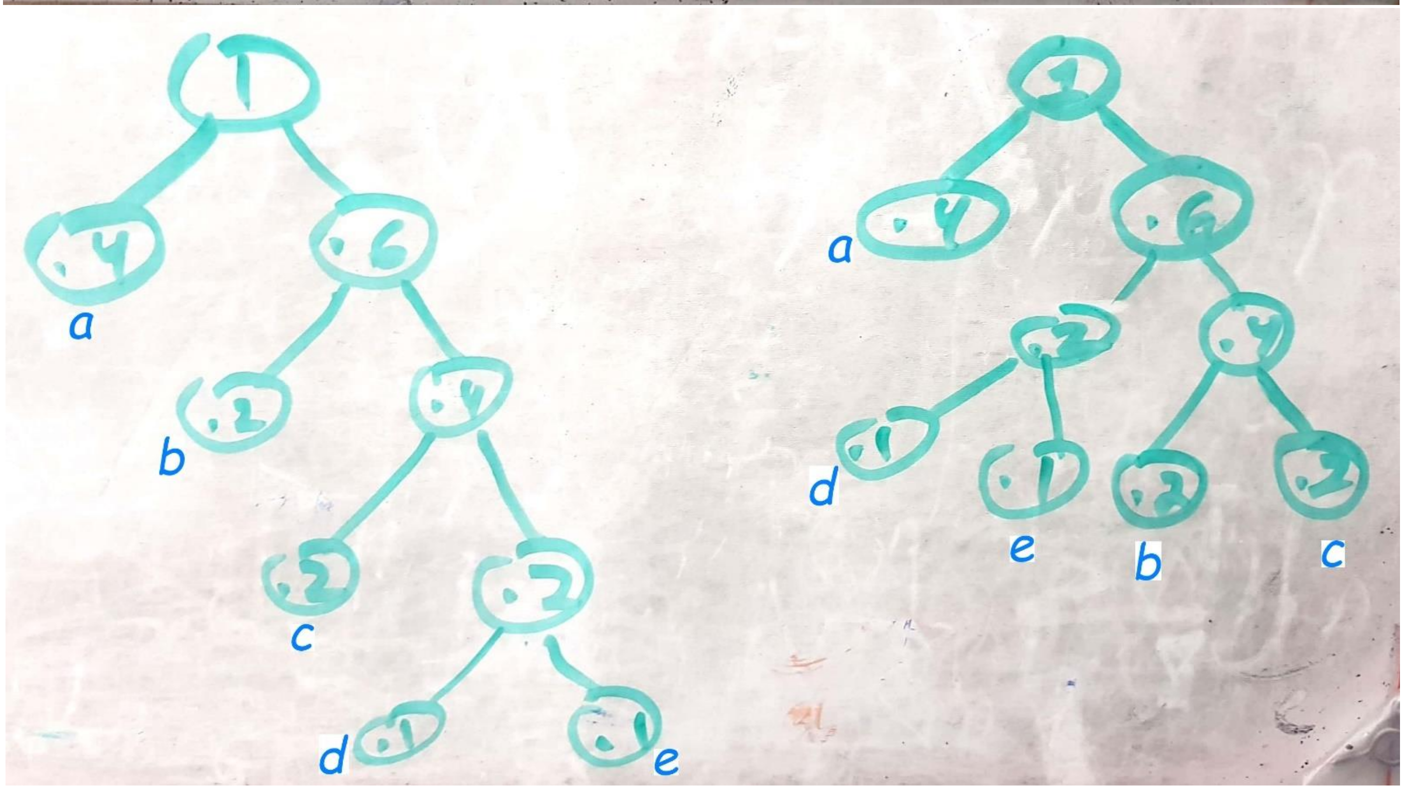
אלגוריתם לקידוד מילה (מספר) σ באמצעות C_δ: $C_\gamma(|\langle \sigma \rangle_2|) \circ Binary^*(\sigma)$.

פתרון מבחן לדוגמה

שאלה 1

סעיף א'





סעיף ב'

נסמן ב-C_1, C_2, C_3 את הקידודים לעיל כאשר הסדר מלמעלה למטה ומשמאל לימין. מספיק לחשב פעם אחת כי כולם שווים אומנם אנו ננצל זאת על מנת לוודא כי לא שגינו.

$$E[C_1] \coloneqq \sum_{i=1}^{n} p_i \ell_i = 0.1 \cdot 3 \cdot 2 + 0.2 \cdot 2 + 0.2 \cdot 2 + 0.4 \cdot 2 = 2.2[b/s]$$
$$E[C_2] \coloneqq \sum_{i=1}^{n} p_i \ell_i = 0.4 \cdot 1 + 0.2 \cdot 2 + 0.2 \cdot 3 + 2 \cdot (0.1 \cdot 4) = 2.2[b/s]$$
$$E[C_3] \coloneqq \sum_{i=1}^{n} p_i \ell_i = 1 \cdot 0.4 + 2 \cdot (3 \cdot 0.1) + 2 \cdot (2 \cdot 0.3) = 2.2[b/s]$$

ואכן מתקבל כי $E[C_1] = E[C_2] = E[C_3]$, כפי שצפינו. חשוב לא לשכוח את יחידות המידה ביט למחרוזת.

סעיף ג'

$$H(P) = -\sum_{i=1}^{n} p_i \log_2 p_i = -\big(0.4 \log_2 0.4 + 2 \cdot (0.2 \cdot \log_2 0.2) + 2 \cdot (0.1 \cdot \log_2 0.1)\big)$$
$$\approx 2.12[b/s]$$

נראה שתוצאה זאת אכן הגיונית שכן $E[C] \geq H(P)$ ואכן $2.2 \geq 2.12$. בדיקה נוספת שרצוי לבצע היא שהפמן נותן מילת קוד ממוצעת שרחוקה לכל היותר בביט מהאנטרופיה ואכן מתקיים כי $2.2 \leq 3.12$.

סעיף ד'

נזכיר שלכל קידוד כנ"ל אנחנו נצטרך להוסיף את ה-prelude ובאלגוריתם הקנוני הוא היה מילת הקוד הראשונה בכל בלוק ואורכה בכל בלוק. ולכן לצמצם את התקורה ככל הניתן וזה יקרה כאשר נקטין את מספר הבלוקים. תשובה נוספת הא שבעץ ה-skeleton המצומצם אנחנו רוצים שההפרש בין העצים השלמים יהיה לכל היותר אחד, הקידוד היחיד שיקיים זאת הוא קידוד C_1.

שאלה 2

סעיף א'

$$\Pr[baccf]=\Pr[b]\cdot\Pr[a]\cdot\Pr[c]\cdot\Pr[c]\cdot\Pr[f]=0.3\cdot 0.2\cdot 0.1^2\cdot 0.1=0.00006=6\cdot 10^{-5}$$

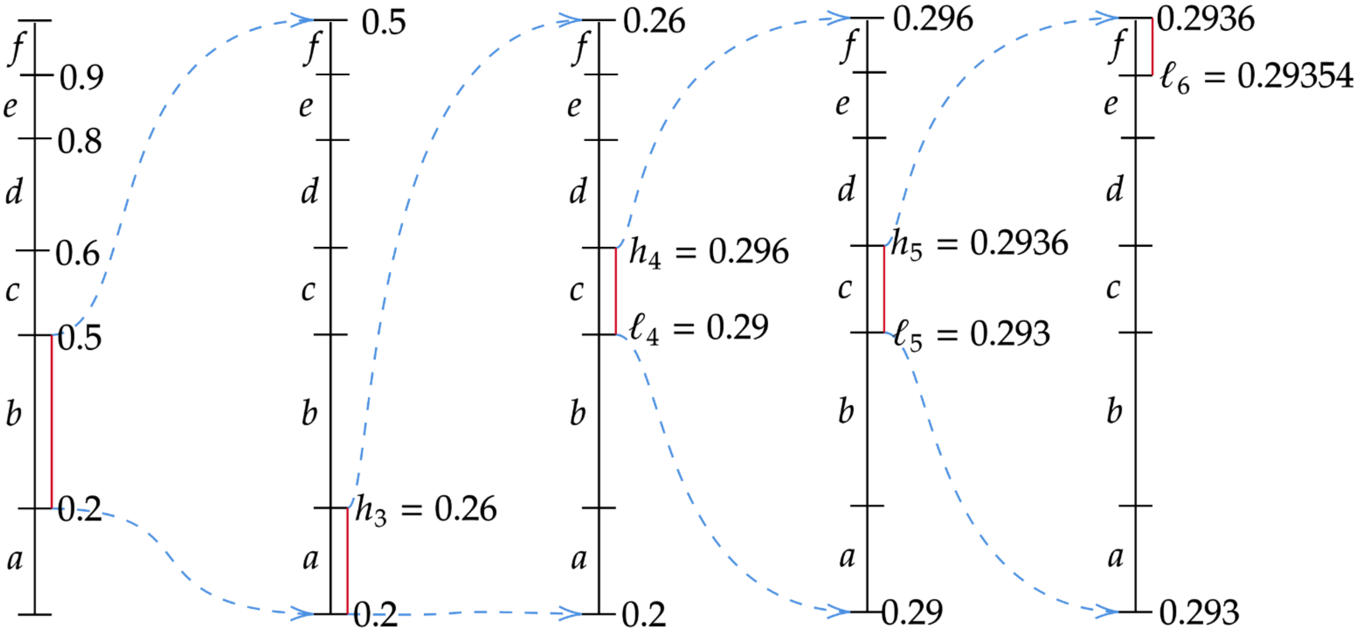
כאשר השוויון הראשון נובע מהאי תלות בין המאורעות.

סעיף ב'

אנחנו מתחילים את ההרצה כאשר $\ell=0,\hbar=1$. קוראים את התו $\sigma=b$ מעדכנים את הטווח להיות $r=1$ ומחשבים

σ	$hb(\sigma)$	$lb(\sigma)$
a	0.2	**0**
b	0.5	**0.2**
c	0.6	**0.5**
d	0.8	**0.6**
e	0.9	**0.8**
f	1	**0.9**

כעת ניתן לקודד באופן הבא:



$$h_3=0.2+0.2\cdot 0.3=0.26 \qquad \ell_5=0.29+0.5\cdot 0.006=0.293$$

$$\ell_4=0.2+0.5\cdot 0.06=0.29 \qquad h_5=0.29+0.6\cdot 0.006=0.2936$$

$$h_4=0.26+0.6\cdot 0.06=0.296 \qquad \ell_6=0.293+0.9\cdot 0.0006=0.29354$$

אנחנו צריכים להוציא מספר out כך ש-$\ell_6\le out<h_6$, נבחר לדוגמה $out=0.29355$.

סעיף ג'

באופן כללי בקידוד אריתמטי הרעיון הוא לעשות scaling כך שבכל פעם הקידוד של $\sigma=\sigma_1\ldots\sigma_n$ הוא באינטרוול המתאים ל-$\prod_{i=1}^{n}\Pr[\sigma_i]$, ואכן $r_6=h_6-\ell_6=0.2936-0.29354=6\cdot 10^{-5}$.

שאלה 3

סעיף א'

נזכיר כי מודל סטטי למחצה אנו שולחים בתקורה (prelude) את מספר התווים את הקידוד של התווים ב-$ascii$. סה"כ נקבל:

$$prelude(C) = 8 + 8 \cdot |\Sigma| = 32[bits]$$

סעיף ב'

כעת נצטרך לקודד גם את ההסתברויות ע"פ נתוני השאלה ניתן לייצג הסתברות באמצעות 4 סיביות ולכן נקבל

$$prelude(C') = 32 + 4 \cdot 3 = 44[bits]$$

סעיף ג'

נזכיר ש-C_γ מורכב משני חלקים: חלק ראשון - $Unary(|Binary(i)|)$, חלק שני - $Binary^*(i)$.נסמן ב-$w = ababbac$ ונשים לב שמתקיים

i	σ_i	C_γ	ℓ_i	$f_i(w)$
1	a	$\langle\|B_1\|\rangle_1 B_1^* = \langle\|1\|\rangle_1 \epsilon = 0$	1	3
2	b	$\langle\|B_2\|\rangle_1 B_2^* = \langle\|10\|\rangle_1 0 = 100$	3	3
3	c	$\langle\|B_3\|\rangle_1 B_3^* = \langle\|11\|\rangle_1 1 = 101$	3	1

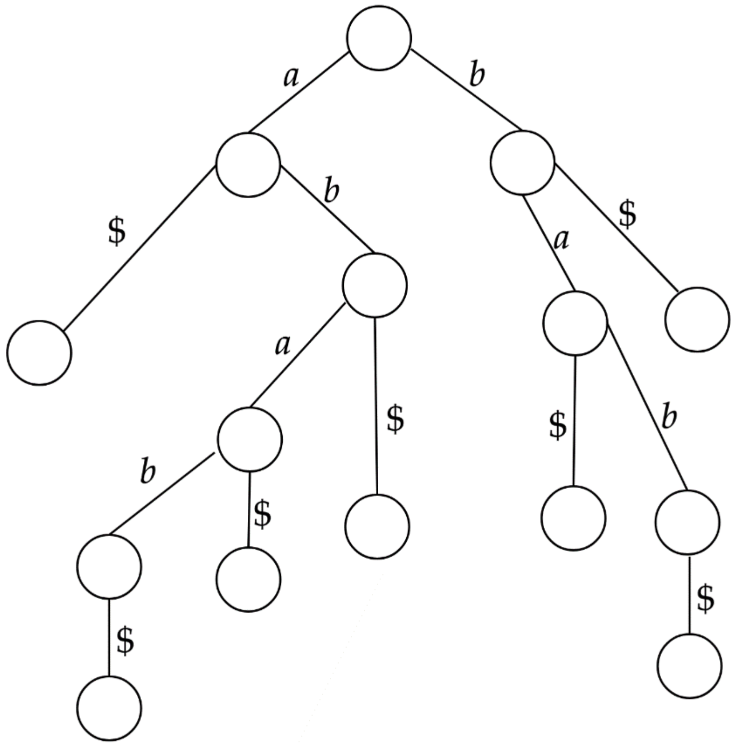
לפיכך מתקיים:

$$E[C,P] = \frac{prelude(C) + \sum_{i=1}^{n} \ell_i \cdot f_i(w)}{7} = \frac{32 + 3 \cdot 1 + 3 \cdot 3 + 1 \cdot 3}{7} = \frac{47}{7} \approx 6.714$$

$$E[C',P] = \frac{prelude(C) + \sum_{i=1}^{n} \ell_i \cdot f_i(w)}{7} = \frac{44 + 3 \cdot 1 + 3 \cdot 3 + 1 \cdot 3}{7} = \frac{59}{7} \approx 8.428$$

שאלה 4

סעיף א'



המילים יכנסו למילון בסדר הבא

$a, b, ab, ba, aba, abab, bab, baba, ababa, ababab, babab, bababa, \dots$

כלומר ברגע המילה האחרונה הופכת להיות הראשונה וממשיכים לפי המחרוזת $\dots ababab$ מהמקום המתאים. האבחנה החשובה לסעיף הבא היא שבכל מילה מאורך i יש 2 מופעים.

סעיף ב'

רוצים שנחשב את

$$\ell := \lim_{n\to\infty} \frac{|LZW(T)|}{|T|}$$

נסמן ב-k את אורך המילה המקסימלית במילון, ונשים לב שאורך הקידוד במקרה זה הוא סכום האורכים של המילים במילון, נגיד בדוגמה ברגע שרשמנו את ab במילון אז פלטנו אותו ואז צריך להמשיך לקודד את כל $n-2$ ה-ab הנוספים. סה"כ סכום האורכים של ההודעות אמור להיות אסמפטוטית כ-$2n$, כלומר נרצה לפתור מה ה-k עבורו מתקיים

$$\sum_{i=1}^{k} 2i = \Theta(2n)$$

היות והסכום הוא סכום חשבוני הוא בערך k^2 ולכן נקבל כי $k = \Theta\left(\sqrt{n}\right)$. היות ויש $2k$ מילים

$$\lim_{n\to\infty} \frac{|LZW(T)|}{|T|} = \lim_{n\to\infty} \frac{2\sqrt{n}}{2n} = 0$$

סעיף ג'

הפמן היה מקודד את $\langle a\rangle = 0, \langle b\rangle = 1$, והיינו משתמשים ב-$2n$ סיביות. כלומר סיבית לכל תו שזה כמו האנטרופיה $H(P) = 1[bps]$.

סעיף ד'

באמצעות $LZSS$ מנקודד את ab ואז נבצע הצבעה עצמית $(2,2n-2)$. בסדר גודל נקבל $O(1)$. אם אי אפשר להשתמש בהצבעות עצמיות אז נבצע $ab(2,2)(4,4)(8,8),\ldots$ כלומר קידוד בסדר גודל $O(\log n)$.

שאלה 5

סעיף א'

נסמן $C_1=\{101,1101,1011,1100\}$ ונחפש סופיות מתנדנדות נשים לב ש-101 היא תחילית של 1011 ולכן 1 היא סיפא מתנדנדת, נוסיף אותה למילון ונקבל $C_2=\{101,1101,1011,1100,1\}$ כעת נבחין כי 1 היא תחילית של 101,1101,1011,1100 ולכן 01,101,011,100 הם סיפות מתנדנדות. היות ו-101 היא מילת קוד, כלומר $101\in C_1$ אזי הקוד אינו UD. נראה כיצד לבנות את הדוגמה:

$$\underbrace{101}_{a}\,\underbrace{1101}_{b}=\underbrace{1011}_{b}\,\underbrace{101}_{a}$$

בעצם לקחנו את המילה המתנגשת הוספנו לה את הסיפא המתנדנדת והשלמנו למילה השניה.

סעיף ב'

תנאי הכרחי על מנת שקוד יהיה UD הוא ש-$K(C)\le 1$. בנוסף אנו יודעים ש-$\ell_1=\ell_2=\ell_3=3$ וכל שאר $\ell_4=\cdots=\ell_{x+3}=8$, לפיכך נקבל את האילוץ:

$$\sum_{i=1}^{x+3}2^{-\ell_i}=3\cdot 2^{-3}+(x+3-4+1)\cdot 2^{-8}\le 1\iff x\le 160\iff \boxed{x+3\le 163}$$

ולכן גודל הא"ב המקסימלי הוא 163.

מבחן 2015 מועד א'

שאלה 1

סעיף א'

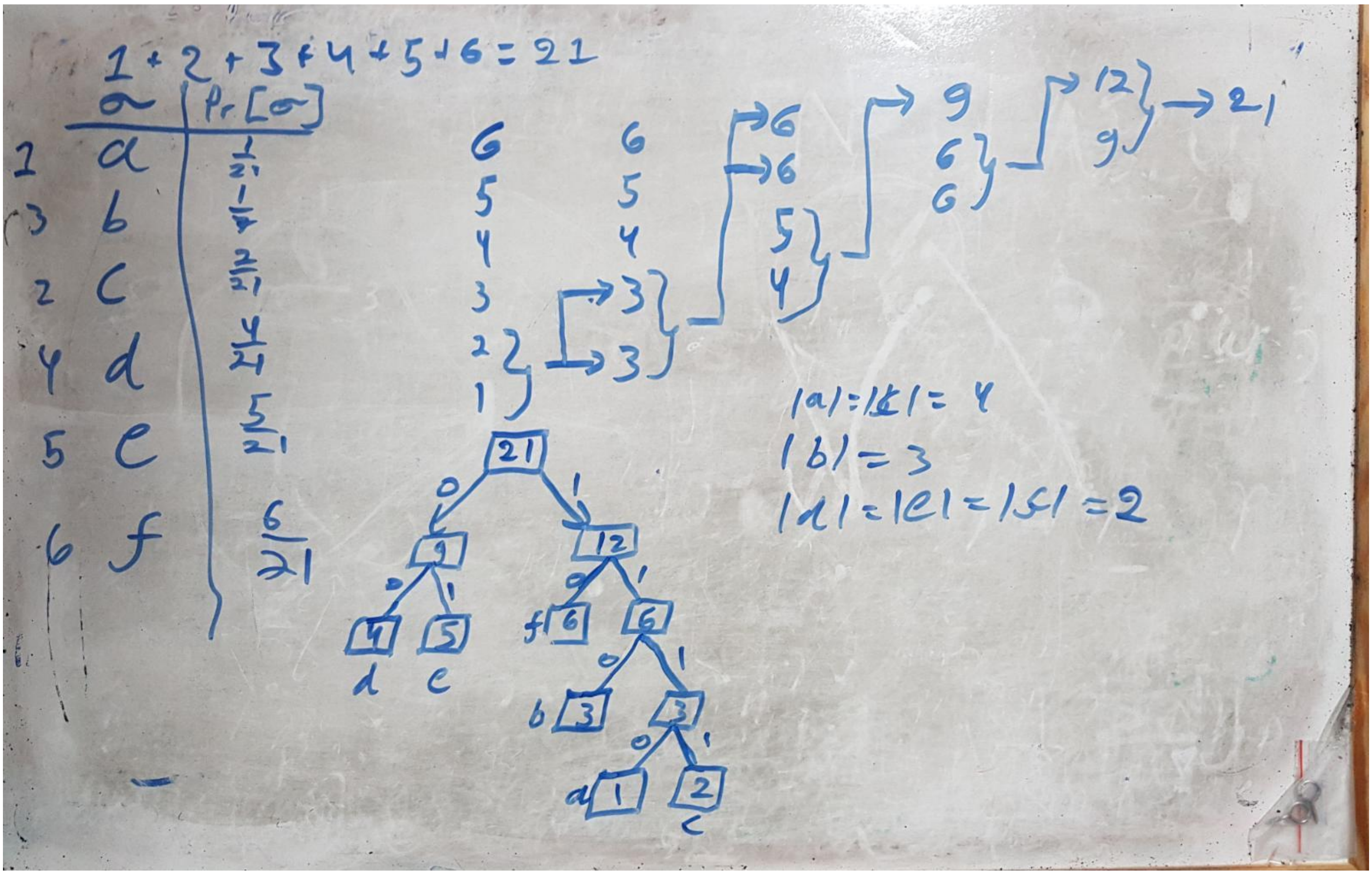
ע"פ משפט קיים קוד פרפיקסי בעל האורכים $1,2,3,3,4$ אם ורק אם

$$2^{-1}+2^{-2}+2^{-3}+2^{-3}+2^{-4}\le 1$$

היות ואגף שמאל שווה ל-$\frac{17}{16}>1$, האי שוויון אינו מתקיים ומכך נובע שלא קיים קוד פרפיקסי כנ"ל.

סעיף ב'

הוכחנו בכיתה כי קוד הפמן הוא קוד אופטימלי לכן נחשב את קוד מילת הקוד הממוצעת בקידוד של הופמן ונראה אם הם שווים:



$$E[C_H,P]=4\cdot\left(\frac{1}{21}+\frac{2}{21}\right)+3\cdot\frac{3}{21}+2\cdot\left(\frac{4+5+6}{21}\right)=\frac{17}{7}\approx 2.428$$

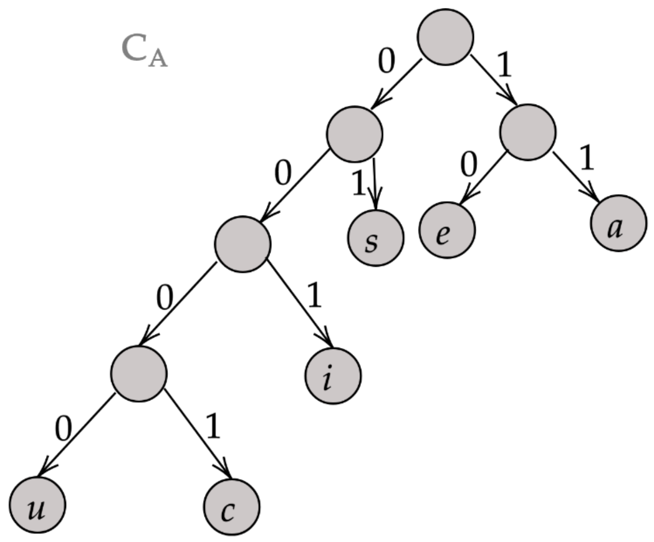
כאשר C_H הוא הקוד ההתקבל מהפמן המתואר בציור.

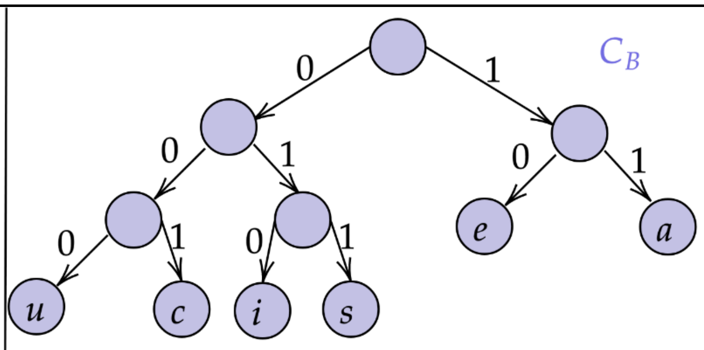
$$E[C,P]=3\cdot\left(\frac{1+3+2+4}{21}\right)+2\cdot\left(\frac{5+6}{21}\right)=\frac{52}{21}\approx 2.476$$

כאשר C הינו הקוד המופיע בשאלה, לפיכך נסיק כי הקוד C אינו אופטימלי, שכן מצאנו קוד אופטימלי יותר.

שאלה 2

סעיף א' + סעיף ב'





$$n = 50 + 15 + 45 + 20 + 10 + X = 140 + X$$

$$\mathbb{E}[C_A] = 2 \cdot \left(\frac{45+50+X}{n}\right) + 3 \cdot \frac{20}{n} + 4 \cdot \left(\frac{15+10}{n}\right) = \frac{2X+350}{n}$$

$$\mathbb{E}[C_B] = 3 \cdot \left(\frac{15+20+X+10}{n}\right) + 2 \cdot \left(\frac{50+45}{n}\right) = \frac{3X+325}{n}$$

היות ושני הקודים התקבלו מהפמן שניהם אופטימלית ולכן $E[C_A] = E[C_B]$ ומכך יתקבל כי $X = 25$.

סעיף ג'

$$\underbrace{11}_{a}\,\underbrace{001}_{c}\,\underbrace{001}_{c}\,\underbrace{10}_{e}\,\underbrace{011}_{s}\,\underbrace{011}_{s}$$

שאלה 3

סעיף א'

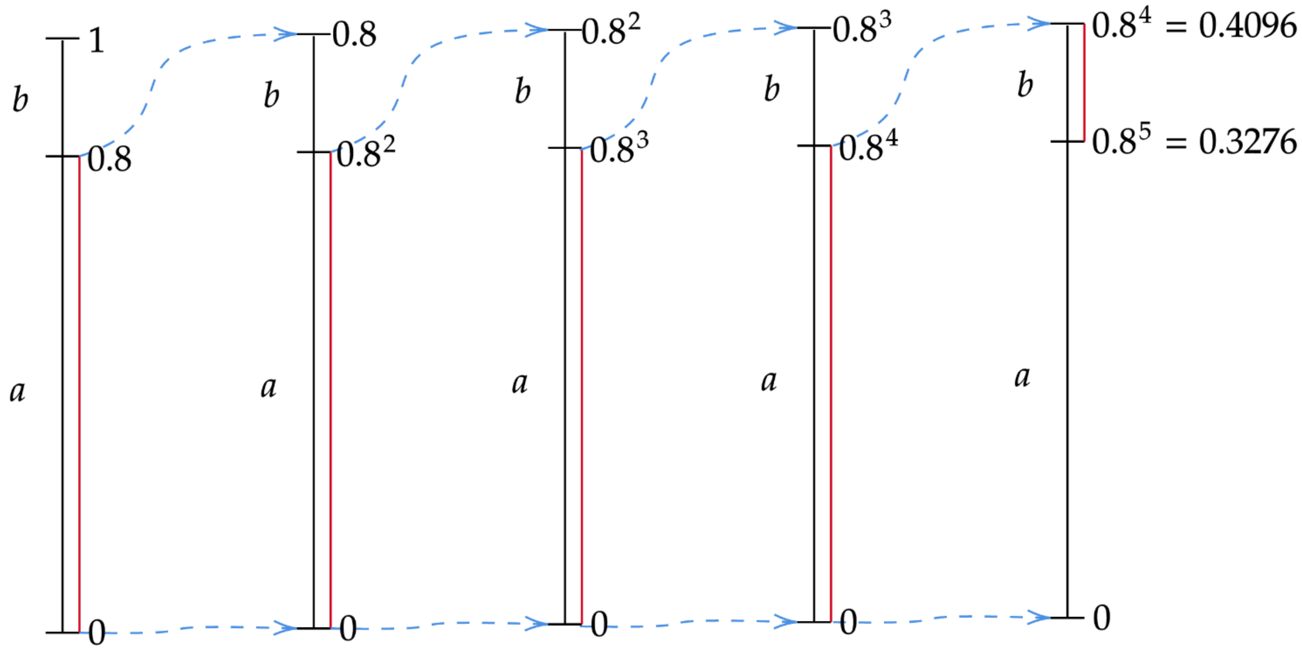
$$H(P) = -\sum_{i=1}^{2} p_i \log_2 p_i = -(0.8\log_2 0.8 + 0.2\log_2 0.2) \approx 0.721[bps]$$

כיוון שישנם 5 אותיות נובע שצריך כ-$5 \cdot H(P) \approx 3.609[bit]$.

סעיף ב' + סעיף ג'

היות וגם ככה בסעיף ג' צריך לקודד את המחרוזת ניתן לקודד אותה כבר משלב זה. אולם על מנת לענות על כוונת המחברת נשים לב שהקוד האדפטיבי נותן לנו את קידוד באורך האינפורמציה של הקטע האחרון היות והקטע האחרון באורך של מכפלת ההסתברויות נקבל כי

$$I = -\lceil \log_2 0.8^4 \cdot 0.2 \rceil = \lceil 3.609 \rceil = 4$$



היות והפלט out צריך לקיים $0.8^5 \le out \le 0.8^4$ נבחר לדוגמה $out = 0.011$ שזה 0.375 ביצוג עשרוני, כלומר 4 סיביות, כפי שקיבלנו מהאינפורמציה (נקודה לא סופרים, מסתבר).

שאלה 4

סעיף א'

אני טעיתי כי חשבתי שזה מספר אחד, אנא שימו לב שהמספר הנ"ל יכול לקודד רצף של תווים אחד אחרי השני.

שאלה 4: א.

0111011111010

1 15 6.

נשים לב שקוד אוניר תמיד נגמר ב-0, לכן נשבור כל פעם שנראה 0 וזה מספר התווים שנצטרך לקחת.

- 0 ← $1_{cr}=0$
- 1110 ← $4_1=1110$ ← נקרא 3 תווים - $1\cdot111$ (4-1)
  $2^3+2^2+2^1+2^0=8+4+2+1=15$ (1 מובן)
- 110 ← $3_1=110$ ← נקרא 2 תווים · $1\,10$ (3-1)
  $2^2+2^1+0=4+2=6$ (1 מובן)

← לכן הפענוח הוא- $\boxed{1\ 15\ 6}$

סעיף ב'

נרחיב את השאלה וניתן דוגמה שבה $|C_\delta(n)| \le |C_\gamma(n)|$ ולהפך. העניין הוא שככל שהמספר גדול יותר ככה C_δ טובה יותר מ-C_γ, ננצל עובדה זאת נתחיל מהמקרה הראשון

$$n = 2^{12} - 1 \Longrightarrow \langle n\rangle_2 = 1^{12} \Longrightarrow C_\gamma(n) = Unary\big(|\langle 12\rangle_2|\big)B^*(12) = 1110100$$
$$C_\delta(n) = C_\gamma\big(|\langle n\rangle_2|\big) \circ B^*(n) = 1110100 \circ 1^{11} \Longrightarrow |C_\delta(n)| = 11 + 7 = 18$$
$$C_\gamma(n) = Unary\big(|\langle n\rangle_2|\big) \circ B^*(n) = 1^{11}0 \circ 1^{11} \Longrightarrow |C_\gamma(n)| = 12 + 11 = 23$$

כעת נענה על המקרה השני:

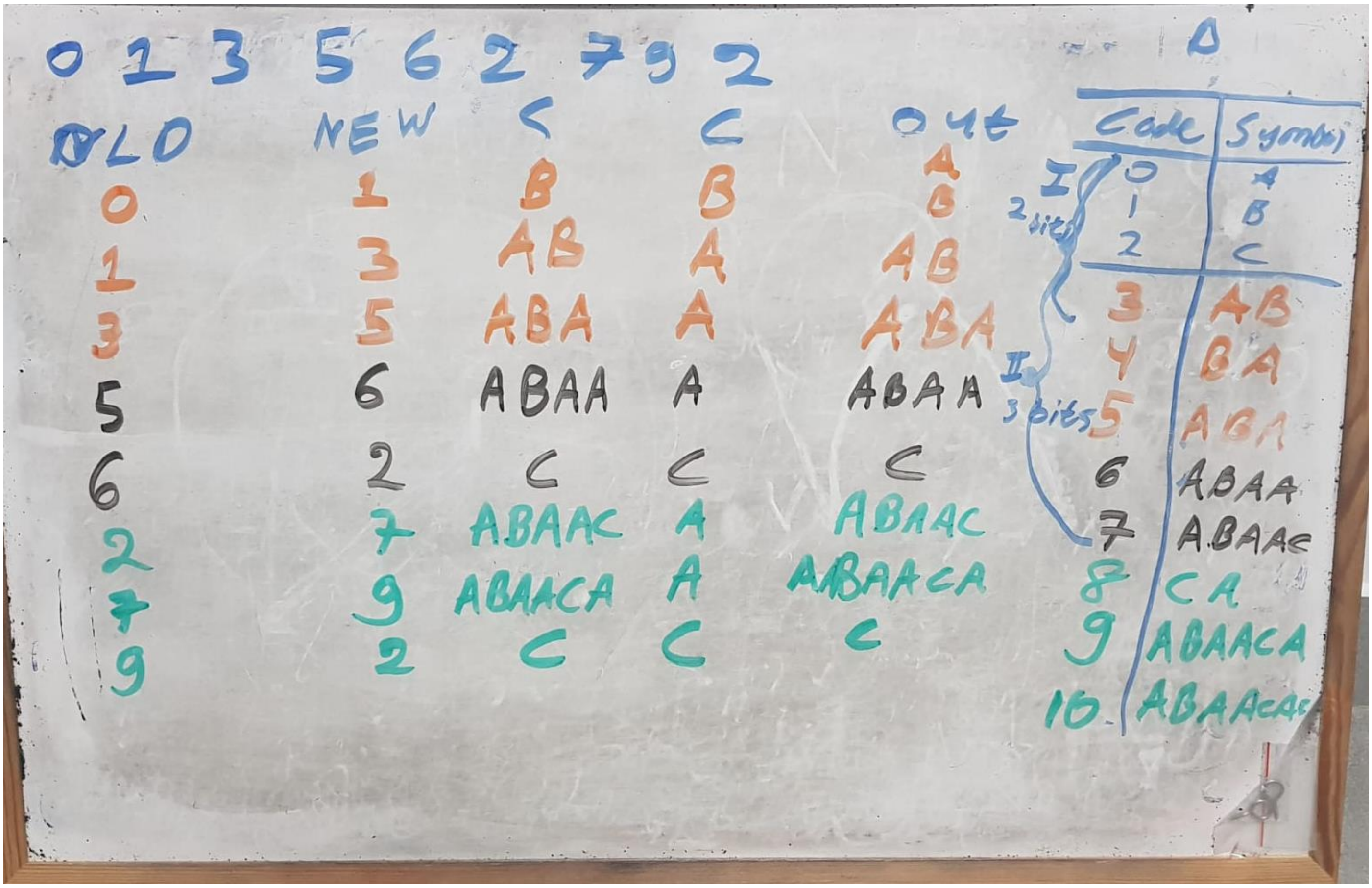
$$n' = 2 \Longrightarrow \langle 2'\rangle_2 = 10 \Longrightarrow C_\gamma(2) = 10 \circ 0$$
$$\Longrightarrow C_\delta(n') = C_\gamma\big(|\langle n'\rangle_2|\big) \circ B^*(n') = C_\gamma(2) \circ 0 = 100 \circ 0$$

הטענה נובעת.

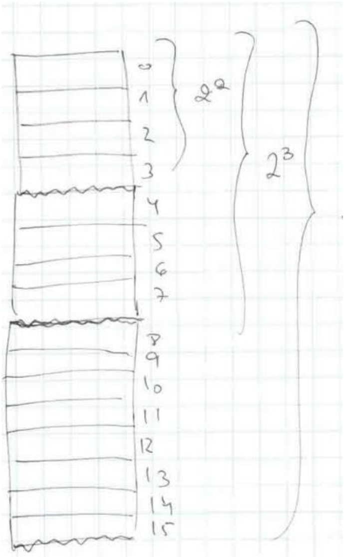
שאלה 5

סעיף א'

נתאר את הריצה באמצעות טבלת מעקב



סעיף ב'



פתרון אחד הוא לשים לב שהמילון מוכפל רק כאשר אנחנו באים לכתוב את הכניסה ב-2^i, והמילון של המקודד והמפענח זהים. לפיכך, כאשר נכתוב את $0,1,3$ השתמשנו בשני סיביות, באופן דומה $5,6,2$ נרשמים באמצעות שלושה סיביות, וכן $7,9,2$ נרשמים באמצעות 4 סיביות. נסכום ונקבל:

$$3 \cdot 2 + 3 \cdot 3 + 3 \cdot 4 = 27bits$$

פתרון הנכון יהיה לקודד שוב את ההודעה

$$ABABABAABAACABAACABAACAC$$

ולספור את מספר הסיביות שנדרשים לקידוד.

codeword	σ
0	***A***
1	***B***
2	***C***
3	***AB***
4	***BA***
5	***ABA***
6	***ABAA***
7	***ABAAC***
8	***CA***
9	***ABAACA***

w	***k***	***Output***	***Enter***	**Bits**
A	*B*	0	*AB*	**2**
B	*A*	1	*BA*	**2** $(\star)$
A	*B*			
AB	*A*	3	*ABA*	**3**
A	*B*			
AB	*A*			
ABA	*A*	5	*ABAA*	**3**
A	*B*			
AB	*A*			
ABA	*A*			
ABAA	*C*	6	*ABAAC*	**3**
C	*A*	2	*CA*	**3**
A	*B*			
AB	*A*			
ABA	*A*			
ABAA	*C*			
ABAAC	*A*	7	*ABAACA*	**4**
A	*B*			
AB	*A*			
ABA	*A*			
ABAA	*C*			
ABAAC	*A*			
ABAACA	*C*	9	*ABAACAC*	**4**
C	*EOF*	2		**4**

ואכן הקידוד הוא $0,1,3,4,6,2,7,9,2$, כאשר נסכום את העמודות של הביטים נקבל שנצטרך

$$2 + 2 + 3 + 3 + 3 + 3 + 4 + 4 + 4 = 28bits$$

נשים לב שבשלב $(\star)$ קודם מוציאים את הקוד ורק אחר כך מכניסים את המילה למילון, ולכן ההגדלה של המילון מתבצעת רק לאחר שלב שרשמנו את 3 בשתי סיביות כלומר 11.

יש פער של סיבית בין התשובות, כרגע עוד לא מצאתי טעות וככל הנראה הפער נובע מכך שברגע שהמפענח מפענח את שלוש הוא עוד לא הכניס את *BA* בניגוד למקודד.

מבחן 2017 מועד ב' – שחזור

שאלה 1 – סעיף א' – האם הקוד $\{0,01,110\}$ הוא UD?

פתרון. נריץ את המבחן

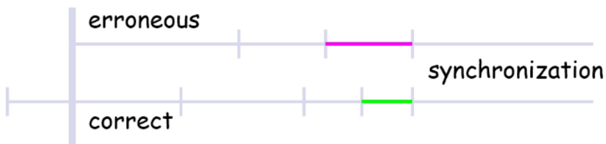
$$\{0,01,110\} \to \{0,01,110,1\} \to \{0,01,110,1,10\} \to \{0,01,110,1,10,0\}$$

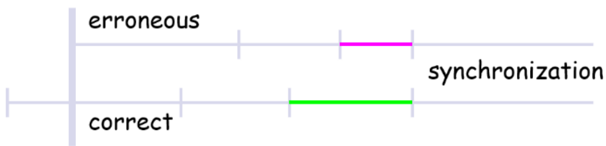
קיבלנו את מילת הקוד 0 ולכן הקוד אינו UD, נראה דוגמה נגדית:

$$0110$$

בדיקות שכדאי לבצע במבחן

- כל התוצאות של קוד הפמן הם אופטימליות כלומר מחזירות מילת קוד ממוצעת מינימלית.
- אם מריצים הפמן מתקיים $E[C,P] \le H(P)+1$.
- $E[C,P] \ge H(P)$.
- לשים לב שעץ skeleton קשור להפמן אז אם שואלים על אחד מהם יעזור לשני.
- בקוד אריתמטי מספר הסיביות שנדרשות הוא $-\log_2\left(\prod_{i=1}^{n} p_i\right)$
- בקידוד LZW קודם כל מוצאים את הפלט ואחר כך מוסיפים את המילה החדשה, כלומר הקידוד שלפני ההכנסה של המילה החדשה נשאר בגודל באותו מספר סיביות שהיה.
- $2|fc[i]+num[i]$, כלומר הערך הדצימאלי של מילת הקוד הראשונה בכל בלוק ועוד מספר המילים בכל בלוק תמיד זוגי.
- דרך לשלול שקוד נתון הינו קוד הפמן היא להראות שהקוד לא שלם, כלומר העץ לא מלא.
- דרך נוספת היא להראות שיש צמתים פנימיים שיש להם מילות קוד.
- עוד דרך היא להסתכל על לשים לב שבגלל שבכל פעם אנחנו ממיינים את השכיחויות אז ישנם עצים שלא יכולים להתקבל, זוהי גם דרך ליצור עץ שאינו יכול להתקבל ע"י הפמן.
- בקידוד בינארי מינימלי לזכור שהמילות הקצרות הם עם השכיחויות הנמכות.
- הפמן ובפרט הפמן קנוני חייב להיות עץ מלא ועץ מלא אם ורק אם $K(C)=1$, ולכן אם נותנים לנו את המערך num של הפמן קנוני, $num=[x_1,\dots,x_n]$ אזי חייב להתקיים
$$K(C)=x_1\cdot 2^{-1}+x_2\cdot 2^{-2}+\cdots x_n\cdot 2^{-n}=1$$
- על מנת לבנות קוד אפיקסי נרשום את כל האופציות למילים בינאריות באורך $1,2,3,4$ ונבחון מה מתאים.
- בהפמן יש סינכרון כי יש מילות קוד שהם סיפא של מילות קוד אחרות וזהו הדרך היחידה שיתבצע סינכרון:





כלומר או שהסיפא של הטעות מסתנכרנת עם הנכון או להפך.

- לכן קוד אפיקסי לא מסתנכרן
- קוד בעל אורך k מסתנכרן רק אם k סיביות נאבדות.

מבחן 2017 מועד ב' שאלה 4: קודד את Mississippi באמצעות הפמן דינמי.