פולינה לוצקר

דחיסה

מבני נתונים ואלגוריתמים

שאלה קריטית

דחיסה

תיאור הבעיה: נרצה להקטין את נפח הנתונים למינימום אפשרי כך שנוכל לקבל חזרה את הנתונים המקוריים.

למה נרצה לדחוס נתונים:

א. הצפנות.

ב. העברת מידע.

ג. לחסוך במקום.

דחיסה

ישנם שני סוגי דחיסה:

- **דחיסה מאבדת מידע (Lossy):** נשתמש במקרים בהם יש אפשרות לאבד מידע. במקרים של מידע חושי כגון תמונה, קול או סרט, ההנחה היא שהחושים האנושיים ממילא אינם מסוגלים להבחין בווריאציות קטנות ולכן לא נורא אם אובד חלק מהמידע, בתנאי, שהמידע שהולך לאיבוד יהיה שולי מבחינת חשיבותו ומבחינת יכולת ההבחנה של התפיסה החושית שלנו.
- **דחיסה משמרת מידע (Lossless):**דחיסה משמרת היא דחיסה המאפשרת לנו להפעיל פונקציה הפוכה לפונקציית הדחיסה, כך שהתוצאה שתתקבל תהיה זהה למקור.

דוגמה-דחיסת שיר

טודו בום/סטטיק ובן אל תבורי

היום חוגגים כמו באמצע הפאבלה,
אצלנו אומרים שבסדר ואצלם אומרים שטודו טודו בום.
היי טודו טודו בום,
היי טודו טודו בום,
הכל בסדר טודו בום, טודו בום, טודו בום.

דוגמה–דחיסת שיר

טודו בום/סטטיק ובן אל תבורי

היום חוגגים כמו באמצע הפאבלה,
אצלנו אומרים שבסדר ואצלם אומרים שטודו טודו בום.
היי טודו טודו בום,
היי טודו טודו בום,
הכל בסדר טודו בום, טודו בום, טודו בום.

במקום לכתוב את כל היצירה, נוכל לקבוע מראש ש:טודו= x ולכתוב את השיר מחדש.

דוגמה-דחיסת שיר

טודו בום/סטטיק ובן אל תבורי

היום חוגגים כמו באמצע הפאבלה,
אצלנו אומרים שבסדר ואצלם אומרים שx x בום.
היי x x בום,
היי x x בום,
הכל בסדר x בום, x בום, x בום.

תרגיל בית -תבדקו בכמה דחסנו.

דוגמה–דחיסת שיר

- בדוגמה הקודמת הפננו למילון שכתבנו מראש.
- נרצה לעבור על הטקסט ולחפש טקסט שחוזר על עצמו. נסמן את הטקסט שחוזר על עצמו בסימון מיוחד. נעבור על הטקסט וכל מופע של הטקסט נחליף בסימן המוסכם. הערה: הסימון חייב להיות שונה מהטקסט.

דוגמה– דחיסת תמונה בשחור לבן

תיאור הבעיה:נרצה לאחסן תמונה בשחור לבן.

נסתכל על רצף של תווים המציין עבור כל פיקסל האם הוא שחור או לבן: אות ל תציין לבן ו–ש תציין שחור.

לדוגמה:

לללללללללללשללללללללשששששללללללללללללללללללללללשללללללללללללללל

האם ואיך אפשר לדחוס?

דוגמה– דחיסת תמונה בשחור לבן

תיאור הבעיה:נרצה לאחסן תמונה בשחור לבן.
נסתכל על רצף של תווים המציין עבור כל פיקסל האם הוא שחור או לבן: אות ל תציין לבן ו–ש תציין שחור.
לדוגמה שורה מהאחסון:

ללללללללללללשללללללללשששששללללללללללללללללללללללשללללללללללללללל

האם ואיך אפשר לדחוס?

נכתוב כמה נקודות שחורות וכמה לבנות יש בכל רצף של צבע. לדוגמה:

16 1 22 5 8 1 12

האם זה מספיק?

דוגמה– דחיסת תמונה בשחור לבן

תיאור הבעיה:נרצה לאחסן תמונה בשחור לבן.
נסתכל על רצף של תווים המציין עבור כל פיקסל האם הוא שחור או לבן: אות ל תציין לבן ו-ש תציין שחור.
לדוגמה שורה מהאחסון:

לללללללללללשללללללללשששששללללללללללללללללללללללשלללללללללללללללל

האם ואיך אפשר לדחוס?

נכתוב כמה נקודות שחורות וכמה לבנות יש בכל רצף של צבע. לדוגמה:

16 1 22 5 8 1 12

האם זה מספיק??לא! כי זה שקול גם לרצף:

ששששששששששששלששששששששללללששששששששששששששששששששששלששששששש
שששששששש

דוגמה- דחיסת תמונה בשחור לבן

תיאור הבעיה:נרצה לאחסן תמונה בשחור לבן.
נסתכל על רצף של תווים המציין עבור כל פיקסל האם הוא שחור או לבן: אות ל תציין לבן ו-ש תציין שחור.
לדוגמה שורה מהאחסון:

ללללללללללללשללללללללשששששללללללללללללללללללללללשלללללללללללללללל

האם ואיך אפשר לדחוס?

נכתוב כמה נקודות שחורות וכמה לבנות יש בכל רצף של צבע. לדוגמה:

12 1 8 5 22 1 16

נוכל תמיד להחליט שהמספר הראשון זה כמות הפיקסלים הלבנים (אם מתחיל בשחור- 0)

או שנוכל לכתוב ליד כל מספר את הסוג של המספר. לדוגמה:

12ל, ש, 8ל, 5ש, 22ל, ש, 16ל (הפסיקים והרווחים הוספו להבהרת הדוגמה) בדוגמה הנ"ל ייצגנו קטע של 65 תווים בעזרת 15- חסכון של 15%!

הערה: לקידוד הנ"ל קוראים קידוד אורך חזרה RLE והדוגמה מבוססת על דוגמה של וויקפדיה.

דוגמה–דחיסה

האם ניתן לדחוס את הקטע

אטויגפ]בצ09׳קWER?

דוגמה-דחיסה

האם ניתן לדחוס את הקטע

אטויגפ]בצ09'קWER?

לא! למה?

דחיסה – הנחות

דחיסת נתונים ניתנת תחת ההנחה כי הרבה מידע חוזר על עצמו או מידע שניתן לייצג בצורה חסכונית יותר.

- בדחיסה נחפש יתִירות בקובץ- דברים שלא נדרשים או שננסה נצמצם חזרות.

העיקרון הבסיסי בדחיסה הוא כי מידע שאנו עושים בו שימוש יש בו "סדר" מסוים.

קובץ אקראי לחלוטין לא ניתן לדחוס!

מידת האי סדר מכונה "אנטרופיה".

אנטרופיה

אנטרופיה: מדד לואריאביליות/חוסר אחידות של התפלגות.(במילים אחרות: מדד לאי ודאות).

אנטרופיה

<u>אנטרופיה:</u> מדד לואריאביליות/חוסר אחידות של התפלגות.

X הוא מ"מ המקבל ערכים $\{x_1, x_2, \dots, x_n\}$ עם הסתברויות $p_1, p_2, \dots, p_3$.

האנרטופיה של X היא:

$$H(X) = -\sum_{i=1}^{n} p_i log\,(p_i)$$

(עבור מ"מ בדיד)

$$H(X) = -\int_{-\infty}^{\infty} f(x)\log(f(x))\mathrm{dx}$$

(עבור מ"מ רציף, כאשר f היא פוקנציית הצפיפות של X).

← האנטרופיה לא תלויה בערכים עצמם, אלא רק בהסתברויות!

הערה: בסיס ה log מוגדר להיות מספר האפשרויות השונות לקבלת הסתברות P_i.

דוגמה

<u>דוגמא:</u> X מקבל את ערכים 0,1 בהסתברויות p_0, p_1.

- אם $p_1 = 1$ ו-$p_0 = 0$ אז נקבל $H(X) = 0$ ← סדר מוחלט, יודעים בוודאות איזה ערך X מקבל.
- אם $p_1 = p_0 = \frac{1}{2}$ (התפלגות אחידה) אז נקבל: $H(X) = -2\left(\frac{1}{2}log\frac{1}{2}\right) = log2$ - זו האנטרופיה המקסימלית: עובר N ערכים אפשריים ש-X יכול לקבל: $H(X) = logN$, כלומר אי אפשר לדעת איזה ערך X מקבל.

← $H(X) \geq 0$ תמיד.

דוגמא

יהי X משתנה מקרי המקבל 1 בהסתברות p ואפס אחרת. אזי האנטרופיה שלו היא $H(X)=-[p\log(p)+(1-p)\log(1-p)]$. במקרה בו p=1/2 ובסיס הלוג הוא 2, נקבל

$$H(X)=-1/2\log_2(1/2)-1/2(\log_2(1/2)=$$
$$=-\log_2(1/2)=1bit$$

תרגיל:

מטילים מטבע הוגן (0/1) 3 פעמים. מה האנטרופיה של מספר ה-1?

תרגיל:

מטילים מטבע הוגן (0/1) 3 פעמים. מה האנטרופיה של מספר ה-1?

פיתרון:

נחשב את ההסתברויות של 3 הטלות – יש 2^3 אפשרויות:

מספר ה-1	0	1	2	3
p_i	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

$$H(X) = -\left(2\frac{1}{8}log\frac{1}{8} + 2\frac{3}{8}log\frac{3}{8}\right) = -\left(\frac{1}{4}log\frac{1}{8} + \frac{3}{4}log\frac{3}{8}\right) = \approx 1.81$$

מרכיבי דחיסה

בכל דחיסה חייבים להיות שני מרכיבים: הדחיסה והפענוח.

קידוד

<u>דוגמא</u> – נניח שיש לנו קובץ עם 100,000 תווים עם אלפבית של 6 אותיות.

- אם נקודד באמצעות קוד שהוא באורך קבוע 0 נצטרך 3 ביטים לייצוג כל תו ← 300,000 ביטים.
- באורך משתנה – ננצל את העובדה שיש אותיות שכיחות יותר , אותן נקודד ע"י קוד קצר יותר, על חשבון אותיות פחות שכיחות.

אותיות	f	e	d	c	b	a
שכיחות (אלפים)	5	9	16	12	13	45
קידוד	1100	1101	111	100	101	0

← מספר ביטים לקידוד הקובץ - 224,000 ביטים (חיסכון של 25% במקום).

פענוח

איך מקודדים ומפענחים קידוד?

- קידוד – שרשור של הקודים.
- פענוח:
  - באורך קבוע – כל n ביטים זה תו.
  - באורך משתנה - נשתמש *בקוד רישות*.

שאלה: מה זה קוד רישות?

פענוח

<u>*איך מקודדים ומפענחים קידוד?*</u>

- קידוד – שרשור של הקודים.
- פענוח:

- באורך קבוע – כל n ביטים זה תו.
- באורך משתנה - נשתמש *בקוד רישות.*

קוד רישות: קוד שבו אין מילה שהיא רישא של מילה אחרת.

פענוח

אותיות	f	e	d	c	b	a
שכיחות (אלפים)	5	9	16	12	13	45
קידוד	1100	1101	111	100	101	0

בדוגמא הקודמת: 10101010

פענוח

אותיות	f	e	d	c	b	a
שכיחות (אלפים)	5	9	16	12	13	45
קידוד	1100	1101	111	100	101	0

בדוגמא הקודמת: 10101010

בודקים האם 1 הוא קידוד – לא. 10 - גם לא. 101 – רק b. ואז 0 – רק a. וכך הלאה.

ייצוג (תרגיל לחימום)

- במחשב כל תו מקודד כמחרוזת של ביטים (0/1).

<u>תרגיל:</u>

הציעו מבנה נתונים התומך בהכנסה, הוספה וחיפוש של מחרוזות ביטים ב-O(k) (k- אורך המחרוזת).

ייצוג (תרגיל לחימום)

- במחשב כל תו מקודד כמחרוזת של ביטים (0/1).

<u>תרגיל:</u>

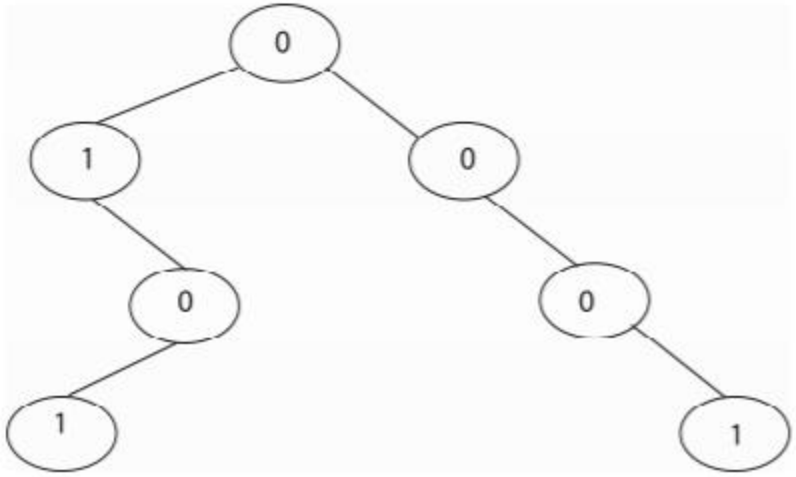
הציעו מבנה נתונים התומך בהכנסה, הוספה וחיפוש של מחרוזות ביטים ב-O(k) (k- אורך המחרוזת).

נשתמש בעץ בינארי. כל צומת יחזיק ערך אינפורמציה 0 או 1. כל מחרוזת תהיה מסלול בעץ – 0=שמאלה, 1=ימינה. אם מסיימים מסלול בעץ בצומת שהערך בו הוא 1 אז המחרוזת נמצאת במבנה.

ייצוג (תרגיל לחימום)

נשתמש בעץ בינארי. כל צומת יחזיק ערך אינפורמציה 0 או 1. כל מחרוזת תהיה מסלול בעץ – 0=שמאלה, 1=ימינה. אם מסיימים מסלול בעץ בצומת שהערך בו הוא 1 אז המחרוזת נמצאת במבנה.

דוגמא – העץ הבא מכיל את המחרוזות הבאות: 0,010, 111.



ייצוג (תרגיל לחימום)

אלגוריתם הוספה: (רקורסיבי)

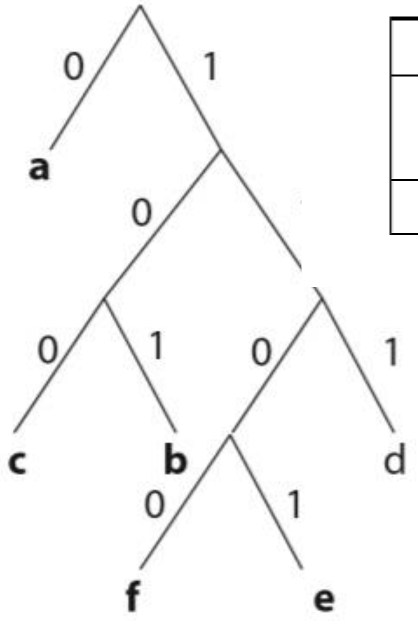
s- מצביע למחרוזת הביטים
n- מספר הביטים שנשאר להוסיף
T- מצביע לצומת בעץ

```
Insert( s, n, T):
If n==0
        valid (T) = 1
else if s==0
        next=Left(T)
        if next==null
                Left(T) = new node
        Insert(s+1, n-1, Left(T))
else  // s==1
        next=Right(T)
        if next==null
                Right(T) = new node
        Insert(s+1, n-1, Right(T))
return
```

סיבוכיות: $T(k) = T(k-1) + O(1) = O(k)$
(הוכחה באינדוקציה).
← בבית – חיפוש והוצאה.

קוד רישות– מבנה

- דרך נוחה לייצג קוד רישות הוא בעץ בינארי, שבו העלים הם התווים, ומילת קידוד היא מסלול מהשורש לעלה. כאשר פונים שמאלה זה 0 וימינה זה 1. (זה לא עץ חיפוש!)



אותיות	f	e	d	c	b	a
שכיחות (אלפים)	5	9	16	12	13	45
קידוד	1100	1101	111	100	101	0

תזכורת

לא ניתן לקודד ערך של מ"מ x בפחות מ-$H(x)$ ביטים במוצע.

לכל אלגוריתם דחיסה, אם קיים טקסט שמתקצר (נדחס), בהכרח שקיים טקסט שמתארך.

הערה: לבחינה תצטרכו לדעת את כל הטענות שלמדתם בנושא.

תזכורת– המשך

נניח שלכל מילה i יש שכיחות בטקסט מסויים P_i (הסתברות למילה מסוימת), כך ש$0 \leq P_i \leq 1$ וכן $\sum_i P_i = 1$. נסמן את אורך כל מילה בl_i, ונרצה ש$L := \sum_i P_i * l_i$, שהוא אורך המילה הממוצע, יהיה מינימלי.

משפט: $H \leq L$

קוד הופמן

קלט: C – קבוצה של **n** תווים.

אלגוריתם חמדן שבונה קוד רישות אופטימלי.

האלגוריתם בונה את העץ **T** המייצג אתה הקוד אופטימלי "מלמטה למעלה".

הוא מתחיל מקבוצה בגודל של **n** עלים ומבצע **n-1** פעולות מיזוג ליצירת העץ הסופי.

קוד הופמן– פסאדו קוד

C – אלפבית, Q – תור קדימויות.

הרעיון – לוקחים בכל איטרציה את 2 התווים בעלי השכיחויות הנמוכות ביותר, מוסיפים לעץ ומכניסים לתור קודקוד פנימי חדש עם שכיחות שהיא סכום השכיחויות שלהם. בסוף מחזירים את השורש של העץ.

```
Huffman(C):
n = |C|
Q.insert(C)
for i=1 to n-1
        new node A
        X = Left(A) = Q.Extract-Min()
        Y = Right(A) = Q.Extract-Min()
        f(A) = f(X) + f(Y)
        Q.Insert(A)
return Q.ExtractMin()
```

לכל תו $c \in C$ הוא עצם בעל שכיחות f[c]

התור קדימויות הוא עם השכיחויות f כמפתח. משמש לזיהוי שני העצמים עם השכיחות הכי נמוכה ביותר כדי למזגם

סיבוכיות: O(nlogn).

תרגיל

נניח שיש התפלגות על אותיות בשפה:

הסתברות	אות
0.13	A
0.14	B
0.15	C
0.16	D
0.17	E
0.18	F
0.07	G

מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו לעומת האופטימום (האנטרופיה)?

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

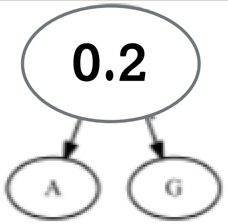
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו לעומת האופטימום (האנטרופיה)?

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A ו-G, ונקבל:

B	C	D	E	F	AG
0.14	0.15	0.16	0.17	0.18	

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

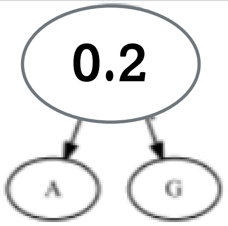
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו לעומת האופטימום (האנטרופיה)?

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A וG, ונקבל:

B	C	D	E	F	AG
0.14	0.15	0.16	0.17	0.18	

כעת, נאחד את B ו C

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

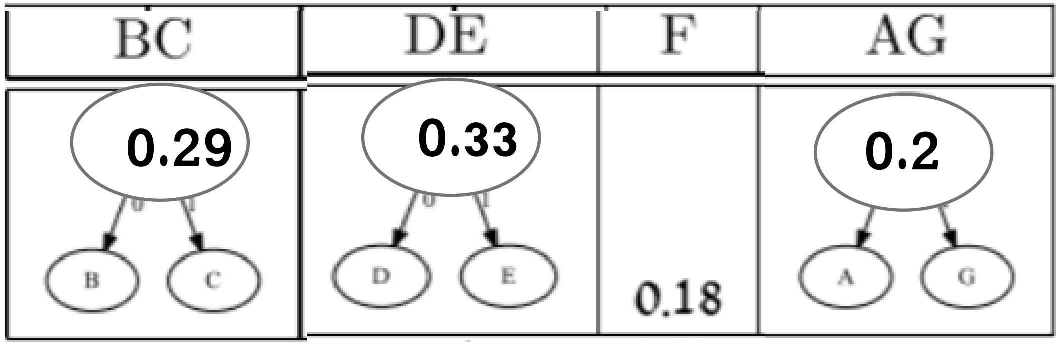
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו לעומת האופטימום (האנטרופיה)?

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A וG, ונקבל:



כעת, נאחד את D ו E

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

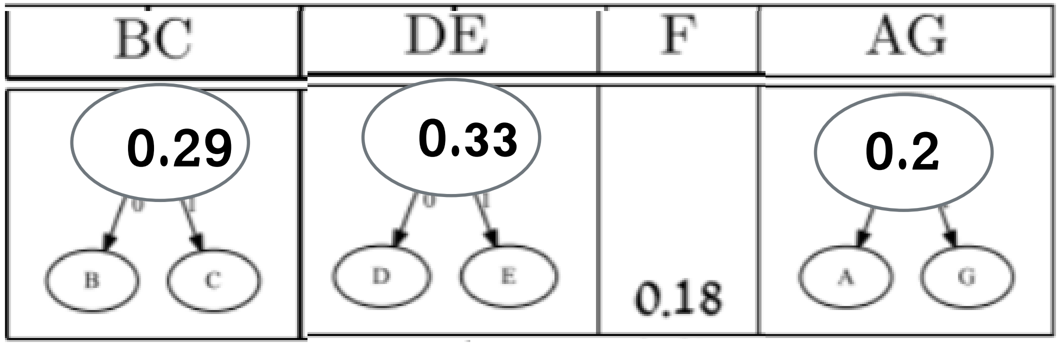
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו לעומת האופטימום (האנטרופיה)?

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A ו-G, ונקבל:



כעת, נאחד את AG וF

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

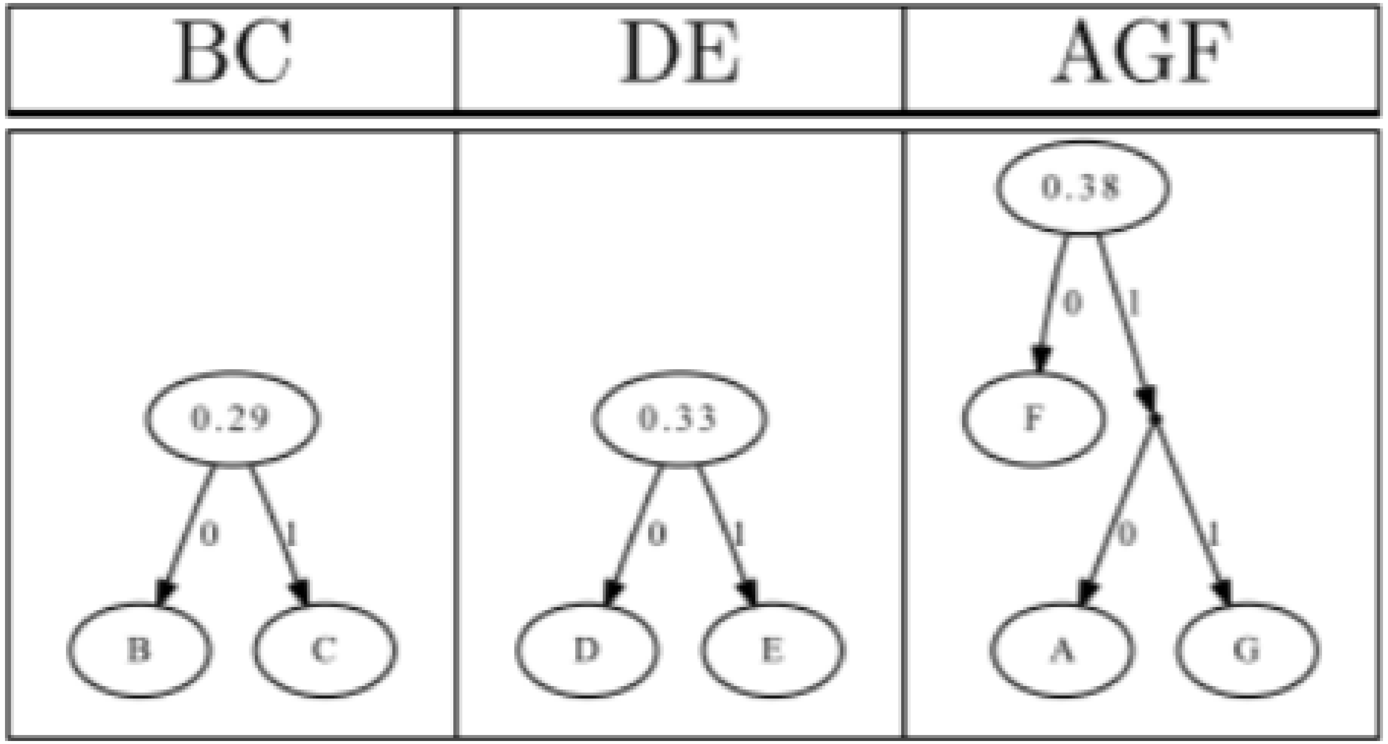
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה של

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A ו-G, ונקבל:



כעת, נאחד את BC וDE

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

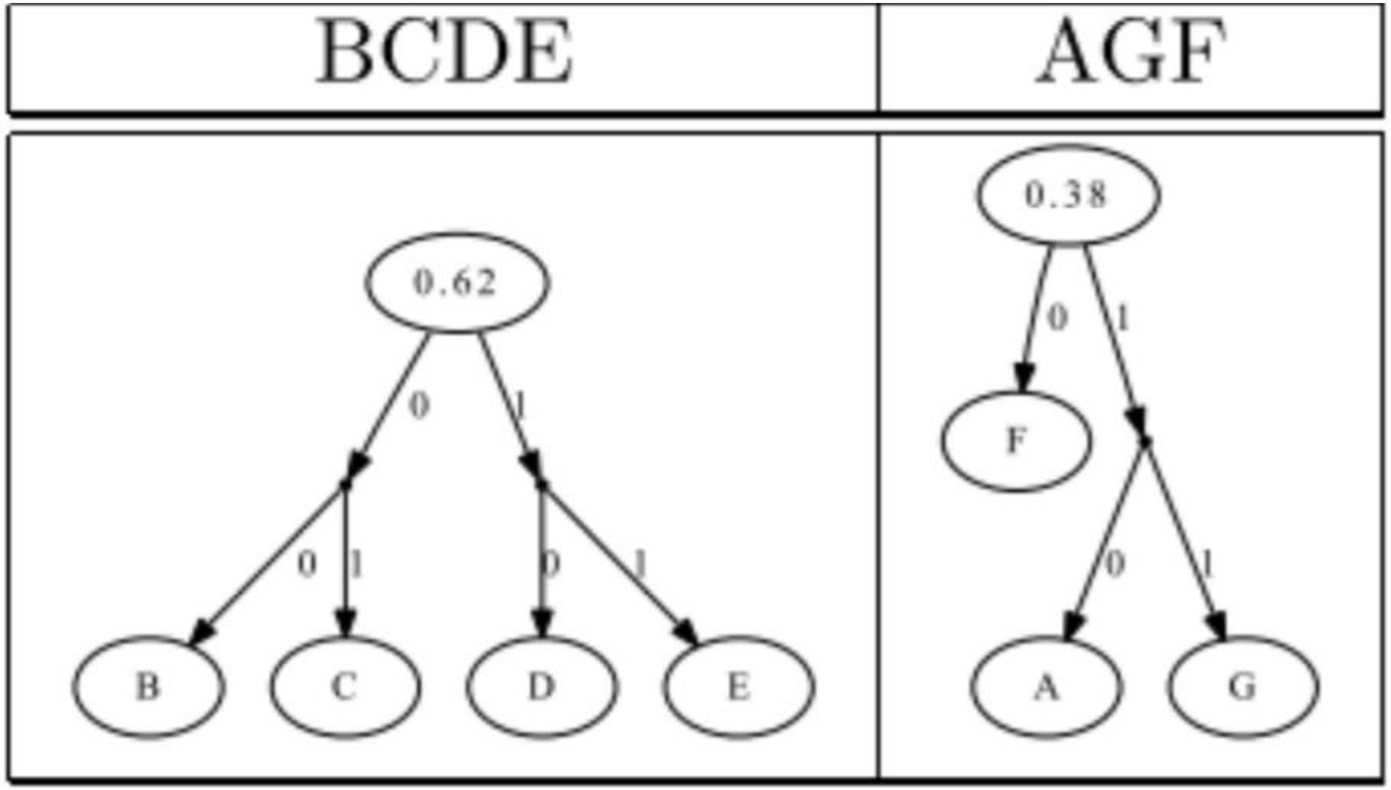
מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו ל

פתרון

נרשום בטבלה:

A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A ו-G, ונקבל:



כעת, נאחד את הסופי ונקבל

תרגיל

נניח שיש התפלגות על אותיות בשפה:

אות	הסתברות
A	0.13
B	0.14
C	0.15
D	0.16
E	0.17
F	0.18
G	0.07

מה עץ הקידוד האופטימלי? מה שיעור הדחיסה שלו ל

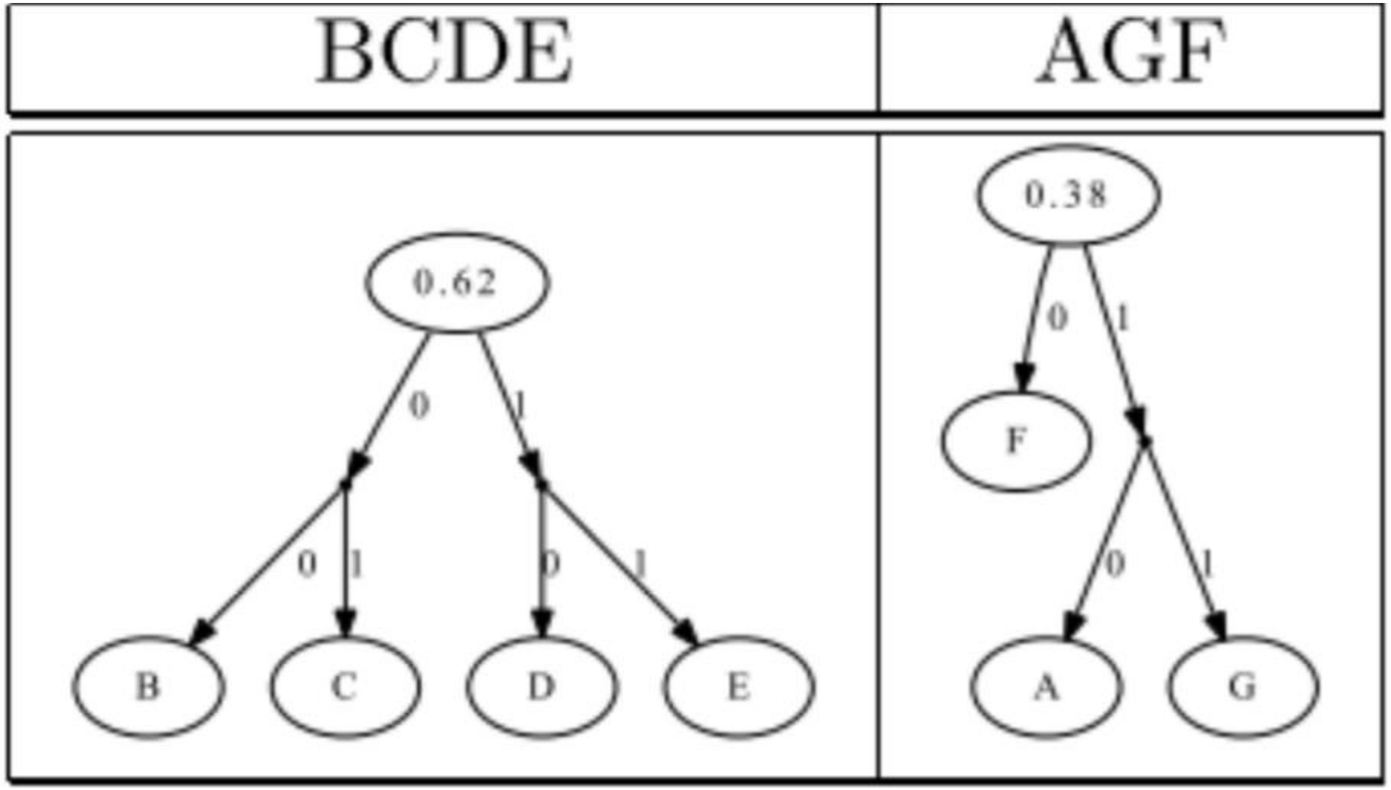
פתרון

נרשום בטבלה:

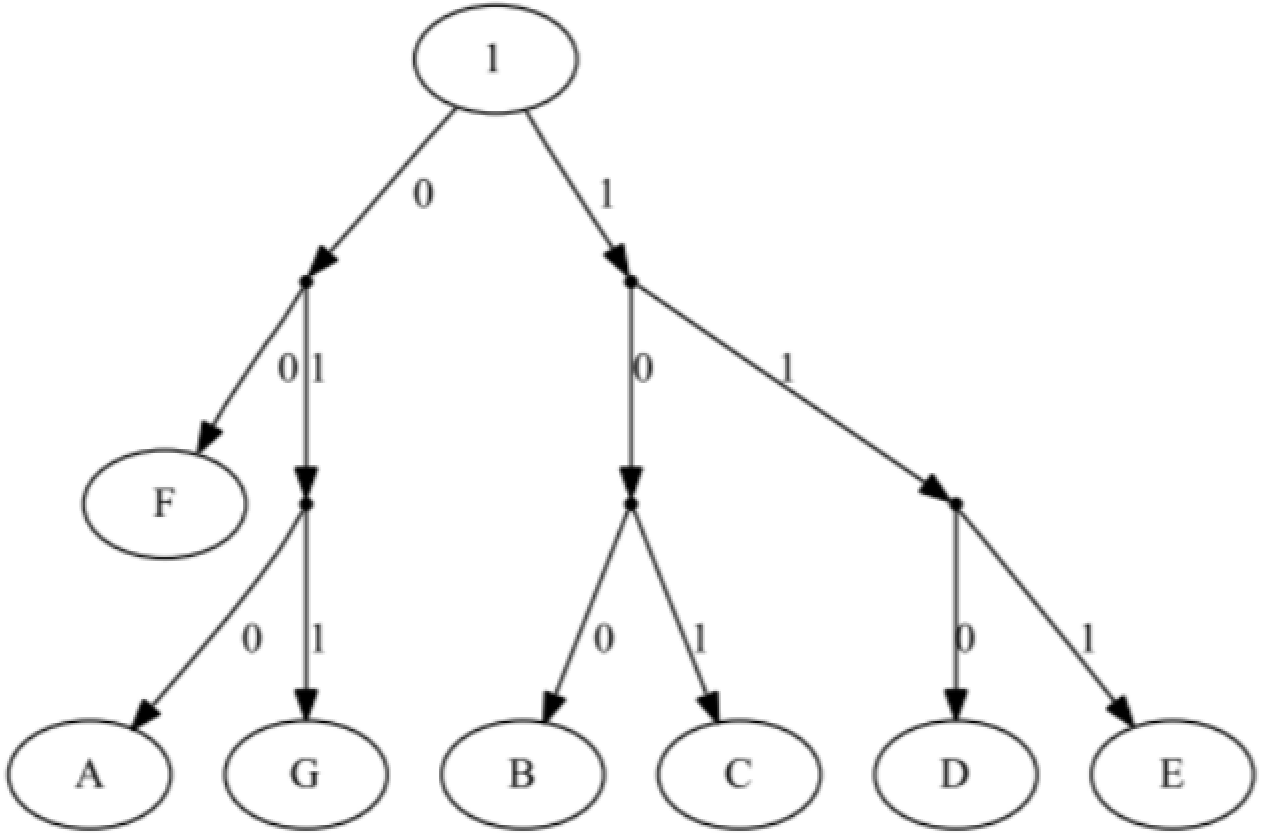
A	B	C	D	E	F	G
0.13	0.14	0.15	0.16	0.17	0.18	0.07

נאחד את הקטנים ביותר, A וG, ונקבל:

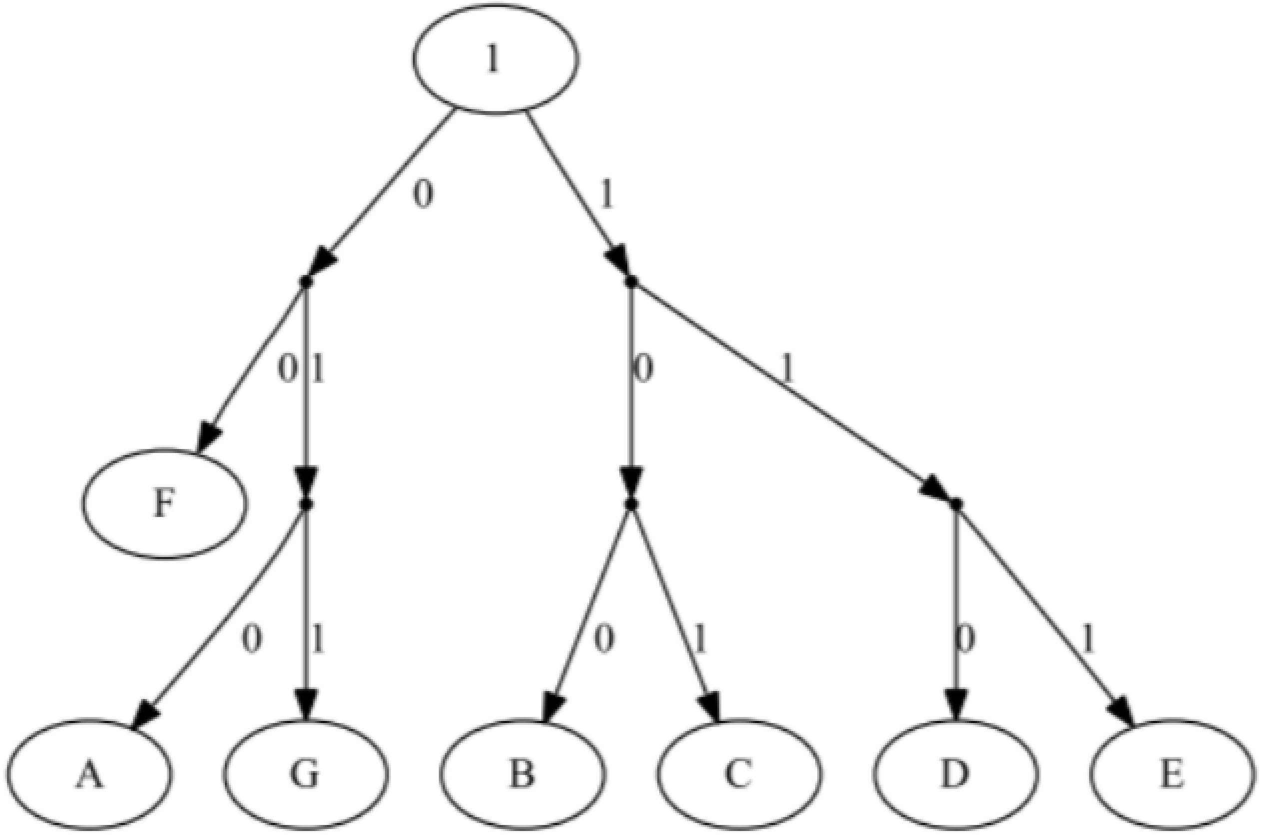
כעת, נאחד את הסופי ונקבל



כעת נחבר את שניהם ביחד ונקבל את העץ הסופי:



כעת נחבר את שניהם ביחד ונקבל את העץ הסופי:



תוחלת כמות הביטית הדרושה לייצוג אות בשפה:

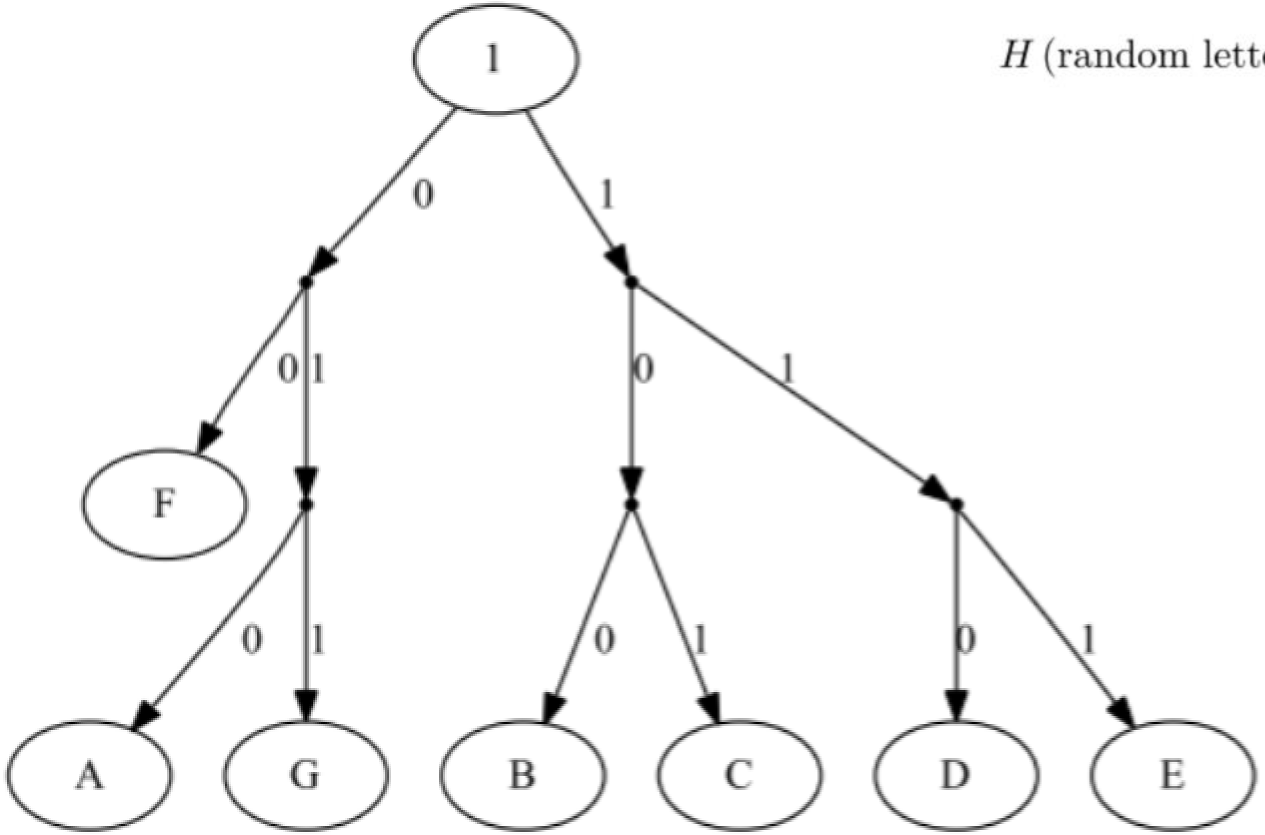
$$\begin{aligned} E(x) &= (0.13+0.14+0.15+0.16+0.17+0.07)\cdot 3 + 0.18\cdot 2 \\ &= 2.82 \end{aligned}$$

כמות הביטים המינימלית הדרושה לייצוג אות בשפה:

$$H(\text{random letter}) = -0.13\lg 0.13 - 0.14\lg 0.14 - \ldots$$

זה יוצא פחות מ2.82 אך במעט.

יוצא 2.76 ביט

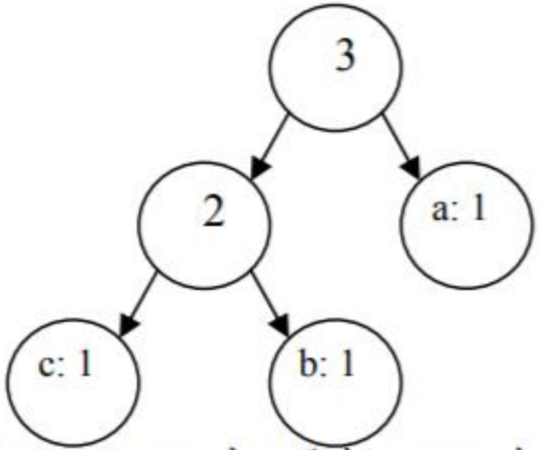


. הוכיחו או הפריכו את הטענה הבאה:

נתון קוד הופמן עבור סדרה של שכיחויות ותווים. יהיו a,b שני תווים כך שאורך מילת הקוד של a קטן ממש מאורך מילת הקוד של b. אז בהכרח השכיחות של a קטנה ממש מהשכיחות של b.

. הוכיחו או הפריכו את הטענה הבאה:
נתון קוד הופמן עבור סדרה של שכיחויות ותווים. יהיו a,b שני תווים כך שאורך מילת הקוד של a קטן ממש מאורך מילת הקוד של b. אז בהכרח השכיחות של a קטנה ממש מהשכיחות של b.

הטענה כמובן אינה נכונה. דרושה דוגמה נגדית.
נתבונן בתווים a: 1, b:1, c:1.
הנה עץ הפמן מתאים:



אכן אורך מילת הקוד של a קטן ממש מאורך מילת הקוד של b, אולם השכיחות של a שווה לזו של b.

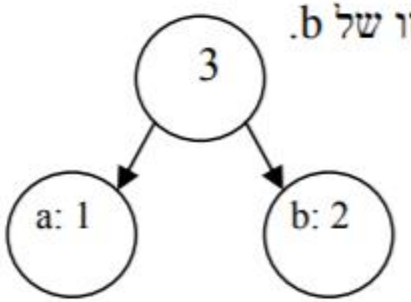
הוכיחו או הפריכו את הטענה הבאה:

נתון קוד הופמן עבור סדרה של שכיחויות ותווים. יהיו a,b שני תווים עם אותו אורך של מילת קוד. אז השכיחות של a שווה לשכיחות של b.

הוכיחו או הפריכו את הטענה הבאה:

נתון קוד הופמן עבור סדרה של שכיחויות ותווים. יהיו a,b שני תווים עם אותו אורך של מילת קוד. אז השכיחות של a שווה לשכיחות של b.

הטענה אינה נכונה. דוגמה נגדית היא a: 1, b:2. בעץ הפמן שמתקבל לשני התווים יש אותו אורך של מילת קוד, אולם השכיחות של a אינה שווה לזו של b.



תרגיל:

1) יהי X מ"מ המקבל ערכים $1,2,...,n$ בהסתברויות $p_1,p_2,...,p_n$ בהתאמה. נניח ש-$2p_i<p_{i+1}$. איך ייראה עץ הקידוד האופטימלי של ערכי X?
2) נניח ש-X מקבל את הערכים $1,2$ בהסתברויות $1/4,\ 3/4$ בהתאמה. מה ניתן לעשות כדי לדחוס ביעילות גדולה יותר סדרת ערכים של X?

פיתרון:

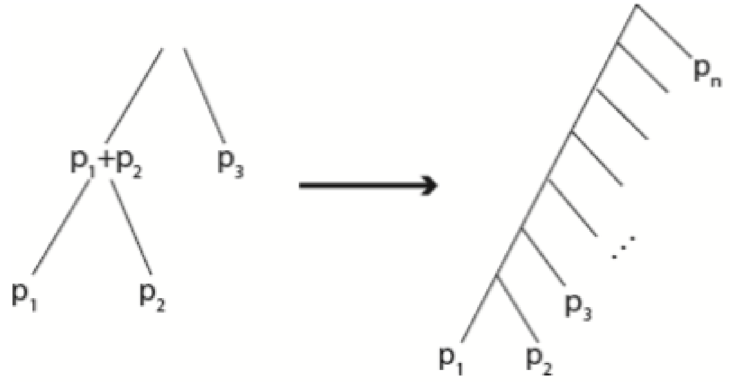
1) נשים לב ש-

$$p_1 + p_2 < 2p_2 + p_2 < 2p_2 < p_3$$

לכן:

$$p_1 + p_2 + \cdots + p_k < p_{k+1}$$

לכן תמיד נאחד את 2 הנמוכים, ואז הנמוך הבא עם הקודקוד החדש שנוצר וכן הלאה:

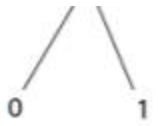


תרגיל:

1) יהי X מ"מ המקבל ערכים $1,2,\dots,n$ בהסתברויות $p_1,p_2,\dots,p_n$ בהתאמה. נניח ש-$2p_i<p_{i+1}$. איך ייראה עץ הקידוד האופטימלי של ערכי X?

2) נניח ש-X מקבל את הערכים $1,2$ בהסתברויות $1/4, 3/4$ בהתאמה. מה ניתן לעשות כדי לדחוס ביעילות גדולה יותר סדרת ערכים של X?

2) העץ האופטימלי לפי הופמן הוא:



כלומר כמות הביטים הממוצעת היא 1.

אבל האנטרופיה נמוכה יותר:

$$H(X) = -\frac{1}{4}log\frac{1}{4}-\frac{3}{4}log\frac{3}{4} = 0.81$$

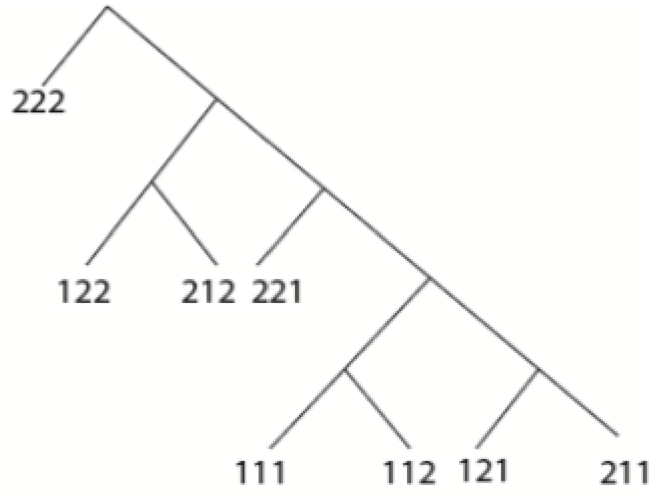
כלומר אפשר לייצג בפחות ביטים (בממוצע).

← כדי לדחוס ביעילות גבוהה יותר,אפשר לדחוס זוגות או שלשות של תווים.

דוגמא – שלשות:

111	112	121	211	122	212	221	222
1/64	3/64	3/64	3/64	9/64	9/64	9/64	27/64

נקבל את העץ הבא:



כמות הביטים הממוצעת של כל שלשה:

$$E(X) = \frac{1}{64}(27 * 1 + 9 * 3 * 3 + 1 * 5 + 3 * 5 * 3) = \frac{158}{64} = 2.46$$

כלומר מקודדים כל שלשה ב-2.46 ביטים, לכן כל תו מקודד ב-0.82 ביטים בממוצע!

למפל-זיו 77

- פותח על ידי אברהם למפל ויעקב זיו מהטכניון.
- שנת 1977.
- האלגוריתם התחיל משפחה של אלגוריתמי דחיסה.
- לא נדרש ידע על התוכן הנדחס.
- משמר מידע.

למפל-זיו 77

האלגוריתם משיג דחיסה על ידי החלפה של מופעים חוזרים של מידע, במצביע לעותק יחיד של אותה פיסת מידע, במופע הראשון שלו בקלט הרצפים הלא דחוס. הרעיון בבסיס הקידוד הוא כי כל מילה בקידוד היא המילה הארוכה ביותר שנראתה עד לאותה נקודת זמן, בתוספת של אות אחת.

למפל-זיו 77

נניח ויש לנו א"ב נתון –כלומר, שהקידוד של הא"ב ידוע.

המטרה: לבנות מילון בזמן אמת.

פסאדו קוד:

1. צור מילון עם כל אותיות הא"ב.
2. מצא את המילה הנוכחית במילון ותרגם אותה ברצף הדחוס.
3. הוסף את המילה הנוכחית + האות שאחריה למילון.
4. המשך כל עוד לא הסתיים העץ.

למפל-זיו 77

דוגמה:

ABABABASABASABABABASASA

מילון התחלתי:

A=0

B=1

S=2

למפל-זיו 77

דוגמה:

A 0
B 1
S 2
AB 3
BA 4
ABA 5
AS 6
SA 7
ABAS 8
SAB 9
BAB 10
BAS 11
SAS 12

ABABASABASABABASASA

0 1 3 0 2 5 7 4 4 7 7

מילון התחלתי:

A=0

B=1

S=2

למפל זיו– מבנה נתונים

הערה: חיפוש אם ערך נמצא במילון בצורת מערך לוקח המון זמן, אז נחזיק אותו בעץ לפי "קוד רישא" רק שהפעם כן מוסיפים לקודקודים פנימיים.

פתיחה

א. צור מילון טריוויאלי עם כל אותיות הא״ב.

ב. תרגם את המילה הראשונה, שהיא אות יחידה ולכן בטוח במילון.

ג. כל עוד לא הגענו לסוף הרצף, נקרא את המילה הנוכחית: אם היא במילון, נתרגם אותה, ואם לא, נתרגם אותה להיות המילה הקודמת + האות הראשונה שלה. נוסיף תרגום זה למילון.

פתיחה-דוגמה

ידוע לנו הקידוד של A=0,B=1,S=2

הוספנו למילון AB=3

A 0
B 1
S 2
AB 3
BA 4
ABA 5
AS 6
SA 7
ABAS 8
SAB 9
BAB 10
BAS 11
SAS 12

0 1 3 0 2 5 7 4 4 7 7

ובעצם נקבל:

ABABASABASABABASASA