תרגיל בית 6

מגיש: תומר מילדוורט | 316081355

2א.

$\{'h': '0', 'g': '10', 'f': '110', 'e': '1110', 'd': '11110', 'c': '111110', 'a': '1111110', 'b': '1111111'\}$

ב.

נסמן: n היא התו ה-n במילון, ולכן היא בעלת התדירות הגדולה ביותר.

מכאן, הקוד האופטימלי הוא מהצורה:

$$\{'n':' 0', \ldots, ' k': ('1' * (k-1)) + '0', \ldots, 'b': '1' * (k-1)\}$$

מכיוון שידוע כי הקורפוס מסודר לפי תדירות של כל תו, ובהתאם לנוסחת הנסיגה של סדרת פיבונאצ'י, אנו יודעים כי זוג התווים בעלי הערך המינימלי יהיו בכל שלב של בניית עץ האפמן שני התווים השמאליים ביותר, מכאן מבנה העץ הוא יחיד. בנוסף, כל צומת תורכב מ-2 בנים, כך שהשמאלי מבניהם הוא עלה והשני צומת בפני עצמו באותה הצורה, פרט לצומת האחרון לו 2 בנים שהם עלים. לפי הגדרת הקידוד, התו בשכיחות הגבוהה ביותר יקבל את הקוד $'0'$. כפי שהראנו קודם, כל תו בשכיחות k הוא בנו של צומת בעומק $k-1$, ומכאן יקודד על ידי $k-1$ פעמים $'1'$ ולבסוף $'0'$. נוסף על כך נשים לב כי האיבר בעל השכיחות הנמוכה ביותר, בדוגמה שלנו התו $'b'$, יהיה בנו **הימני** של צומת בעומק $k-2$ ולכן קידודו יהיה $k-1$ פעמים $'1'$.

ג.

נראה כי:

$$|C(a_1)| - |C(a_n)| = 0$$

ראשית, נתבונן באופן בו נבנה עץ האפמן.

לפי הנתון עבור $n = 256$, $a_1 < a_2 < \cdots < a_n$ וגם $2a_1 > a_n$. לכן, נטען כי חיבור של שני האיברים (עץ או אות) בעלי התדירות המינימלית גדול מכל שאר התדירויות האחרות.

במילים אחרות, לכל $1 \leq k \leq n$ ולכל $k < m \leq n$ מתקיים $a_k + a_{k+1} > a_m$. לכן, עבור $m = 1$ מתקיים: $a_k + a_{k+1} \geq a_1 + a_2 > 2a_1 > a_n$.

ובאופן כללי, לכל $h \leq \log(n)$ ועבור $m \geq k + 2^h$ מתקיים:

$$a_k + \cdots + a_{k+2^h-1} \geq 2^h \cdot a_1 > 2^{h-1} \cdot a_n > a_m + \cdots + a_{m+2^{h-1}-1}$$

לכן, האופן בו יבנה עץ האפמן הוא כזה שבכל פעם יתחברו שני איברים בעלי עומק עץ מינימלי וזהה. אופן חיבור זה יצור לבסוף עץ האפמן במבנה של עץ שלם. בעץ שלם כלל העלים נמצאים בעומק שווה, לכן כמות הקשתות בינם לבין השורש שווה. לפי קידוד האפמן, אורך קוד האפמן המאפיין כל עלה (אות) יהיה באורך שווה, מכאן: $|C(a_n)| - |C(a_1)| = 0$.

ד.

נראה כי $\log 300 \notin \mathbb{N}$. לכן, בשלב מסוים בבניית העץ נקבל רשימה בעלת כמות איברים אי זוגית. מכאן, בהכרח יתבצע חיבור של שני עצים בעלי עומק שונה, ולכן לא יתקבל עץ שלם. יתר על כן, יווצר פער של עומק אחד לפחות בין העלה העמוק ביותר לעלה בעומק הכי קטן.

ניתן למנות את כמות הפעמים שמקרה כמתואר יתרחש, והוא כמות הפעמים בו הספרה 1 מופיעה בייצוג הבינארי של n, פרט לראשון. כמות זו היא מספר הפעמים בה לאחר מחיקת הביט האחרון (חלוקה ב-2) יווצר מספר אי זוגי (מספר בינארי שנגמר ב-1). כשנותר המספר 1 בלבד, נותרנו עם עץ יחיד המהווה את תנאי העצירה של האלגוריתם.

העלה העמוק ביותר ייצג את תדירות a_1. לכן, עבור $n = 300_{10} = 100101100_2$ מתקיים:

$$|C(a_n)| - |C(a_1)| = 3$$

ה.

נוכיח כי: $|C(a_n)| - |C(a_1)| = 5$.

בהנתן התנאים החדשים, נחלק את בניית העץ ל-2 חלקים:

ראשית, בעקבות תנאי b, ראשית יתבצעו חיבורים בין 16 האיברים הראשונים. מתנאי a ומהנתון כי $a_1 < \cdots < a_n$, ראינו בסעיפים הקודמים כי יווצר עץ שלם בעומק 4 ($\log 16 = 4$). נסמנו ב-T.

כעת, קיבלנו רשימה ובה 257 איברים ($272 - 16 + 1 = 257$). נשים לב כי מנתון c נובע אי השוויון הבא:

$T < 16a_{16} < a_{17}$, לכן ב"סיבוב" הבא בבניית העץ יתחברו האיברים T ו-a_{17}, נסמן את חיבורם ב-T'.

כעת, נבנה את העץ השני:

נניח כי $T' > a_n$ מכאן, יתבצעו חיבורים בבניית העץ באופן הבא:

$$T', [a_{18} + a_{19}], \dots, [a_{n-2} + a_{n-1}], a_n$$

לכן, בשלב הבא בבניית העץ יתחברו האיברים T' ו-a_n. מכך, a_n נמצא בגובה זהה לצומת המהווה אבא לשורש של העץ הנבנה קודם לכן. מכאן, עומקו יהיה גדול ב-1 מעומק השורש שתחתיו (T). לכן, פער אורכי קודי ההאפמן של a_1 ו-a_n יהיו העומק של $T + 1$. הראנו כי עומקו של העץ T כשלעצמו הוא 4, לכן:

$$|C(a_n)| - |C(a_1)| = 5$$

נניח כי מתקיים $T' \leq a_n$, לכן קיים $i < n$ עבורו מתקיים $T' > a_i$, לכן בשלב מסוים נקבל (לדוגמה עבור $i > 21$):

$$[T' + a_i], [a_{18} + a_{19}], \dots [a_{i-1} + a_{i-2}], \dots, [a_{n-1} + a_n]$$

ומכאן a_n עדיין יהיה באותו הגובה של T' (יתקבל עץ שלם) ועדיין יתקיים:

$$|C(a_n)| - |C(a_1)| = 5$$

3א.

דוגמה למחרוזת: "$abcxxxxabc$".

פלט הביניים שיתקבל בשתי ההרצות: $[a, b, c, x, [1, 3], [7, 3]]$

ב.

קיימת מחרוזת s המקיימת את הנדרש:

$s\ =\ \text{“}ababbbababababa\text{”}$

$LZW\_compress(s)\ =\ ['a','b','a','b','b', [4, 3], [2, 7]]$

$LZW\_compress\_new(s)\ =\ ['a','b','a','b','b','b','a', [2, 8]]$

לכן:

$len(inter\_to\_bin(LZW\_compress(s)))\ =\ 78$

$len(inter\_to\_bin(LZW\_compress\_new(s)))\ =\ 76$

ומתקיים:

$len(inter\_to\_bin(LZW\_compress(s)))\ >\ len(inter\_to\_bin(LZW\_compress\_new(s)))$

ג.

לא קיימת מחרוזת s המקיימת

$len(inter\_to\_bin(LZW\_compress(s)))\ <\ len(inter\_to\_bin(LZW\_compress\_new(s)))$

הפונקציה $LZW\_compress\_new$ בודקת באופן רקורסיבי, ע"י השוואת כמות הביטים של res1 ו- res2, המהווים חלקים מפלט הביניים, את הדרך בה **כמות הביטים היא הנמוכה ביותר** עבור פלט ביניים.
הפונקציה $LZW\_compress$ בודקת **קיום של פלט ביניים כלשהו** (ע"י חיפוש חזרות בעזרת הפונקציה $maxmatch$) ומבלי לבדוק האם הוא האופטימלי מבחינת כמות הביטים בפלט הביניים הסופי.

לכן, גם אם $LZW\_compress$ תחזיר במקרה את פלט הביניים בו כמות הביטים היא הקצרה ביותר האפשרית, נדע בוודאות כי גם $LZW\_compress\_new$ החזירה פלט ביניים עם כמות ביטים זהה.
מכאן, לא ייתכן:

$len(inter\_to\_bin(LZW\_compress(s)))\ <\ len(inter\_to\_bin(LZW\_compress\_new(s)))$

לכל s.

4א.

(x_1, x_2, x_3)	$(x_1, x_2, x_3, x_1+x_2, x_1+x_3, x_2+x_3, x_1+x_2+x_3)$
$(0,0,0)$	$(0,0,0,0,0,0,0)$
$(0,0,1)$	$(0,0,1,0,1,1,1)$
$(0,1,1)$	$(0,1,1,1,1,0,0)$
$(1,1,1)$	$(1,1,1,0,0,0,1)$

ב.

בקוד הנתון $d = 4$. לדוגמה:

$$w_1 = (0,1,0,1,0,1,1)$$

$$w_2 = (1,0,1,1,0,1,0)$$

לא קיימת מילד קוד במרחק $d < 4$. נתבונן בסכמה של מילת הקוד ונבחין כי כל ביט מופיע ב-4 מופעים שונים: לבדו, פעמיים בחיבור עם כל אחד מהביטים האחרים לחוד, ופעם נוספת בחיבור של כלל הביטים יחד.

בפרט, נשים לב כי לכל זוג ביטים 2 מופעים משותפים בסכמה. לכן, נאמר כי בהנתן מילת קוד $w_n \in \{0,1\}^7$:

שינוי של ביט אחד מתוך x_1, x_2, x_3 יגרור 4 שינויים ב-w_n ולכן $d = 4$.

שינוי של 2 ביטים מתוך x_1, x_2, x_3 יגרור 2 שינויים **עבור כל ביט**. נראה כי בשני המופעים בו שני הביטים מופיעים יחד, הביט יחליף את ערכו פעמיים ולכן למעשה יחזור לערכו המקורי ב-w_n. לכן יתבצעו 4 החלפות בהשוואה ל-w_n ולכן $d = 4$.

שינוי של 3 הביטים מתוך x_1, x_2, x_3 יגרור 4 החלפות גם כן: במקום כל ביט מופיע לבדו יתבצע חילוף, בכל חיבור בין שני ביטים ערך הביט יוחלף פעמיים ולכן יחזור לערכו המקורי, נראה כי כאשר נחבר את שלושת הביטים ערך הביט הסופי יתחלף 3 פעמים ולכן למעשה יחליף את ערכו בהשוואה ל-w_n. לכן $d = 4$.

משמע, לכל 2 מילות קוד מרחק השווה ל-4, ובפרט זהו גם המרחק המינימלי.

ג.

נטען כי הטענה נכונה. נתבונן ב:

$$y = (0,0,0,0,0,0,0)$$

$$w_1 = (0,0,1,0,1,1,1)$$

$$w_2 = (1,1,1,0,0,0,1)$$

בסעיף א׳ הראינו כי אלו שלוש מילות קוד, ולכן המרחק ביניהן הוא $d = 4$ ובפרט המרחק בין w_1, w_2 ל-y שווה, כפי שהראינו בסעיף ב׳. לכן, זהו גם המרחק המינימלי מכל מילת קוד אחרת ל-y, ולכן שני התנאים מתקיימים כנדרש.

ד.

נחלק למקרים.

עבור $x = 2$:

bad_coding הוא קוד מטיפוס $[n = 12, k = 2, d = 8]$.

עבור $x > 2$:

bad_coding הוא קוד מטיפוס $[n = (|x| + 1) \cdot 4, k = |x|, d = 4]$.

5א.

נחלק את פעילות האלגוריתם CYK ל-3 חלקים עיקריים כפי שראינו גם בתרגול:

א. יצירת המטריצה (רשימות מקוננות) מסדר $n \times n$, וכפי שראינו בתרגול זו פעולה בסיבוכיות של $O(n^2)$.

ב. מילוי האלכסון הראשון ע"י קריאה לפונקציה $fill\_length\_1\_cells$ הפועלת בסיבוכיות של $O(n|R|)$.

ג. מילוי שאר החלק הרלוונטי במטריצה.

הראינו כי חלק ג' הוא המשמעותי ביותר בקביעת הסיבוכיות של כלל האלגוריתם בזמן הממוצע. ראשית, נראה כי שתי הלולאות הראשונות פועלות בסיבוכיות של $O(n^2)$ יחד. לכן, נראה כי סיבוכיות הזמן של $fill\_cell$ היא לפחות $c \cdot n|R|$ או גדולה ממנה.

כעת, נראה כי הלולאה הראשונה רצה $j-(i+1)$ פעמים, לכן כאשר המשתנה $length$ (מהלולאה שבתוכה נקרא $fill\_cell$) מקיים $length > \frac{n}{2}$ מתקיים: $j-(i+1) \geq \frac{n}{2} - 1$. נוסף על כך, נשים לב כי הלולאות rhs ו-lhs פועלות בסיבוכיות של $O(|R|)$. מכאן, בכל האיטרציות בהן מתקיים $length > \frac{n}{2}$ מתקיים גם:

$$fill\_cell = O\left(\left(\frac{n}{2} - 1\right) \cdot |R|\right) = O(n|R|)$$

מכיוון שהאורך של $length$ גדול מ-$\frac{n}{2}$ ביותר מ-$\frac{n}{2}$ איטרציות, הסיבוכיות של חלק ג' כולו תהיה לכל הפחות

$$\left(\frac{n}{2}\right)^2 \cdot \left(\frac{n}{2} - 1\right)|R| = O(n^3|R|)$$

הראינו בתרגול כי האלגוריתם CYK חסום ע"י $O(n^3|R|)$, לכן נוכל לומר כי סיבוכיות CYK במקרה הממוצע היא למעשה $\theta(n^3|R|)$ כנדרש.