קידוד ודיחסה

נניח שאנו רוצים לשמור את המחרוזת abacdaaa. זה תופס $8 \times 8 = 64$ ביטים. אבל אם אנחנו יודעים שאנחנו רוצים לשמור רק את ארבעת התווים a, b, c, d, היינו יכולים לייצג אותה ב$2 \times 8 = 16$ ביטים, באמצעות הייצוג $\begin{array}{cc} a & 00 \\ b & 01 \\ c & 10 \\ d & 11 \end{array}$.

אם היו לנו רק 3 תוים - למשל abaccaaa - היינו יכולים להשתמש באורכים שונים כדי לשמור רצפים שונים. ע"י הייצוג $\begin{array}{cc} a & 0 \\ b & 10 \\ c & 11 \end{array}$ היינו יכולים לשמור את המחרוזת ב$2 \cdot 2 + 1 \cdot 2 + 5 \cdot 1 = 11$ ביטים, שכן a תופס רק ביט אחד.

כשאנחנו נותנים את הקודים באורכים שונים, צריך להיזהר שנדע איפה מסתיימת אות אחת ומתחילה אות אחרת.

- נרצה שקוד יהיה בר פענוח יחיד(Unique Decipherability(UD)) כלומר שהקוד של מחרוזת יהיה לו פיענוח אחד ויחיד.
- נרצה שהפענוח יהיה מיידי.

הגדרה

קוד=פונקציה $c : \Sigma \to \{0,1\}^*$(כאשר Σ א"ב)

דוגמה

אם $\begin{array}{cc} a & 01 \\ b & 00 \\ c & 100 \end{array}$ אז $\begin{array}{c} c(a) = 01 \\ c(b) = 00 \\ c(c) = 100 \end{array}$

המשך הגדרה

עבור מחרוזת $S = s_1 s_2 ... s_n$ נסמן $c(S) = c(s_1) \cdot c(s_2) \cdot ... \cdot c(s_n)$(כאשר $\cdot$ מסמן שרשור).

רוצים קוד שלכל S, $c(S)$ בר פענוח יחיד ומיידי.

הגדרה

קוד יקרא קוד תחילי(או קוד רישות) אם לכל $x, y \in \Sigma$, $x \neq y$, $c(x)$ אינו רישא של $c(y)$. כלומר הקידוד של אף אות אינו התחלה של אות אחרת.

הערה

קוד תחילי הוא בר פענוח יחיד ומיידי.

בעיית מציאת קוד תחילי באורך מינימום

קלט $f_1, f_2, ...f_n$ תדירויות של $a_1, a_2, ...a_n \in \Sigma$.

פלט קוד אופטימלי. כלומר קוד $c : \Sigma \to \{0,1\}^*$ כך ש$\sum_{i=1}^{n} f_i l_i$ מינימום, כאשר $l_i = |c(s_i)|$ = האורך של הקוד של a_i.

דוגמה

Σ	f	c
a	99	00
b	99	01
c	99	10
d	99	11

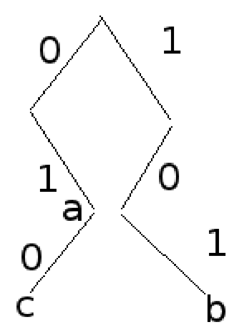
- כל האותיות מופיעות בתדירות שווה, ולכן נייצג את כולן באותו אורך.

אבל אם

Σ	f	c
a	1000	0
b	100	10
c	10	110
d	5	111

- כלומר לאותיות שמופיעות יותר פעמים ניתן קודים יותר קצרים. נשים לב שברגע שקוד אחד הוא 0, אף קוד אחר לא יכול להתחיל ב0, ולכן כל הקודים האחרים חייבים להיות מאורך לפחות 2.

ייצוג קודים ע"י עץ(Trie)



מייצג את

a	01
b	101
c	010

- אם בקודקוד רשומה אות, אז המסלול אליו יהיה הקוד של אותה אות.

אלגוריתם של הופמן(Huffman)

1. נמיין את התדירויות $f_1 \geq f_2 \geq ... \geq f_n$ (המתאימות ל$a_1, ...a_n$)
2. תנאי עצירה: $n = 2$, ואז קוד אחד יקבל 0 והשני יקבל 1
3. אחרת:

* נוציא את f_{n-1} ו-f_n מהרשימה(ובהתאם את a_{n-1} ו-a_n)
* נכניס(למיקום המתאים ברשימה הממויינת) את f_n+f_{n-1}(עם אות חדשה b).

נקבל: $f_1, f_2, ...f_{n-2}, f_{n-1}+f_n$ ובהתאמה $a_1, a_2, ...a_{n-2}, b$

4. בצורה רקורסיבית נבנה עץ עבור $f_1, ...f_{n-2}, f_{n-1}+f_n$($a_1, ...a_{n-2}, b$) ונוציא מהקודקוד b 2 בנים, אחד עבור a_{n-1} ואחד עבור a_n.

זמן ריצת האלגוריתם של הופמן

1. $O(n\log n)$ עבור מיון.

2. מימוש: עבור מערך $O(i)$ לסיבוב ה-i(שכן צריך לדחוף את האיברים כשמכניסים את $f_{n-1}+f_n$). עבור רשימה מקושרת $O(i)$, שכן צריך למצוא את המיקום.
זמן: $O(n+(n-1)+(n-2)+...+2)=O\left(n^2\right)$
עץ מאוזן בינרי: $O(n\log n)$, $\log n$ לכל סיבוב.
בעזרת מחסנית ותור: זמן $O(n)$. נשמור את כל האיברים במחסנית, וכל פעם שמוציאים שני איברים במחסנית - את הסכום שלהם נשמור בתור. האיברים הכי קטנים יהיו או בראש המחסנית(כי זה ממויין) או בתחילת התור(כי שם שמנו את הזוגות הכי קטנים).

סה"כ זה יקח $O(n\log n)$ בגלל שלב 1.

הוכחת נכונות

למה 1

בעץ אופטימלי(=עץ עבורו $\sum f_il_i$ מינימום) לכל קודקוד פנימי יש 2 בנים

הוכחה

נניח בשלילה שקיים אץ אופטימלי T עבורו יש קודקוד פנימי, u, עם בן יחיד, v. נסמן את האבא של u ב-x, וניצור עץ T' מ-T באופן הבא: נמחק את u ו-2 הקשתות המחוברות אליו וניצור קשת חדשה (x,v). T' עץ המתאים לקוד תחילי(כי T מתאים לקוד תחילי).
אבל, $\forall_{u\in T'} l'_u\leq l_u$, ו-$l'_u<l_u$ לכל u בתת-עץ של x, ולכן $\Sigma f_il'_i<\sum f_il_i$ בסתירה לאופטימליות T. מש"ל.

מסקנה

בעץ אופטימלי יש לפחות 2 קודקודים ברמה התחתונה.

הוכחה

אחרת הקודקוד היחיד u ברמה התחתונה מחייב של-x, האבא של u, יהיה בן יחיד. סתירה ללמה 1.

למה 2

בעץ אופטימלי f_{n-1} ו-f_n נמצאים ברמה התחתונה.

הוכחה

נניח בשלילה שהטענה לא נכונה, בה"כ נניח שf_{n-1} אינו ברמה התחתונה. מהמסקנה נסיק שברמה התחתונה יש $f_i < f_{n-1}$. מהעץ T האופטימלי ניצור T' ע"י החלפת f_i וf_{i-1}.

נסמן $M = \sum f_i l_i$ (עבור T) ו$M' = \sum f_i l_i'$.

$$M' = M + f_{n-1}\left(l_i - l_{i-1}\right) - f_i\left(l_i - l_{n-1}\right) = M + \left(f_{i-1} - f_i\right)\left(l_i - l_{n-1}\right) < M$$

למה 3

קיים עץ אופטימלי עבורו f_{n-1} (a_{n-1}) וf_n (a_n) אחים.

טענה

עץ הופמן אופטימלי, כלומר מחזיר עץ עבורו $\sum f_i l_i$ מינימום.

הוכחה

באינדוקצה על n (מספר התדירויות)

בסיס האינדוקציה: $n = 2$. הופמן יוצר את העץ היחיד שאפשר.

נניח נכונות ל$n-1$ ונוכיח לn. יהי T_1 עץ אופטימלי עבור $f_1, \ldots f_n$ (ל$a_1, \ldots a_n$) המקיים את למה 3, כלומר f_{n-1} וf_n אחים.

נבנה T_2 מT_1 ע"י מחיקת העלים $f_n - 1$ וf_n ונסמן את האבא בb (עם $f_{n-1} + f_n$)

למה

T_2 אופטימלי (ל$f_1, \ldots f_{n-2}, f_{n-1} + f_n$)

הוכחה נניח בשלילה שT_2 אינו אופטימלי. לכן קיים T_3 אופטימלי ל$f_1, \ldots f_{n-2}, f_{n-1}+$ f_n שונה מT_2. ניקח את T_3 ונבנה ממנו T_4 ע"י הוצאת 2 בנים מb (המתאים ל$f_{n-1} + f_n$). בן אחד יסומן עם f_{n-1} והאחר עם f_n. T_4 עץ תחילי עבור $f_1, \ldots f_n$.

נסמן $M_1 = \sum f_i l_i$ בT_1

$M_2 = \sum f_i' l_i'$ בT_2

$M_3 = \sum f_i'' l_i''$ בT_3

$M_4 = \sum f_i''' l_i'''$ בT_4

$$M_1 = M_2 + f_{n-1} + f_n$$

$$M_4 = M_3 + f_{n-1} + f_n$$

$$M_1 = M_2 + f_{n-1} + f_n > M_3 + f_{n-1} + f_n = M_4$$

קיבלנו ...
מש"ל למה.

כאשר בונים עץ הופמן עבור $f_1, ...f_n$ בשלב (3) האלגוריתם מצמצם את f_{n-1} וf_n לקודקוד b עם $f_{n-1} + f_n$ וברקורסיה בונה עץ T' עבור התדירויות $f_1, ...f_{n-1} + f_{n-2}$.
לפי הנחת האינדוקציה T' אופטימלית(עבור $f_1, ...f_{n-1} + f_n$)

$$M_{T_2} = M_{T'}$$

בצעד האחרון הופמן מקבל עץ T עם עלות M

$$M = M_{T'} + f_{n-1} + f_n = M_{T_2} + f_{n-1} + f_n = M_{T_1}$$

לכן T אופטימלי(כי T_1 אופטימלי). מש"ל