קידודים - קוד האפמן

הבעיה

קלט: קובץ עם תוים כל תו מופיע בתדירות מסויימת בקובץ.

פתרון חוקי: קידוד של כל תו למחרוזת בינארית כך שניתן לפענח את הקוד בצורה יחידה.

ערך: הערך של פתרון חוקי הוא אורך הקובץ המקודד.

פלט: פתרון חוקי בעל ערך מינימלי.

אפשרות ראשונה: קידוד באורך קבוע:

כל תו ייוצג ע"י מחרוזת בינארית שווה באורך קבוע i. כאשר אם יש n תווים אזי i השלם המינימאלי שמקיים $n \le 2^i$ כלומר $i = \lceil \log n \rceil$.

דוגמא:

קווים	a	b	c	d	e	f
מס' מופעים בקובץ באלפים	45	13	12	16	9	5
קידוד קבוע	000	001	010	011	100	101

אורך הקובץ:

$$3(45+13+12+16+9+5)\cdot 1000 = 300{,}000$$

אפשרות שניה: קידוד באורך משתנה

אורך הקובץ:

קווים	a	b	c	d	e	f
מס' מופעים בקובץ באלפים	45	13	12	16	9	5
אפשרות 2	0	101	100	111	1101	1100

$$1000\cdot(45\cdot 1+13\cdot 3+12\cdot 3+16\cdot 3+9\cdot 3+5\cdot 4) = 224{,}000$$

לא כל קוד באורך משתנה ניתן לפענח

קודים חסרי רישא

הגדרה

קוד יקרא קוד חסר רישא אם אין מילת קוד שהיא רישא של מילת קוד אחרת.

קודים חסרי רישא ניתנים לפיענוח בדרך פשוטה ויחידה.

ניתן להוכיח שלכל שיטת קידוד אחרת ניתן להתאים קוד חסר רישא שיעיל באותה מידה

*כל קוד טוב ניתן להפוך לקוד חסר רישא טוב כמוהו, לכן נתמקד רק בקודים כאלו.

ייצוג קוד חסר רישא ע"י עצים בינאריים

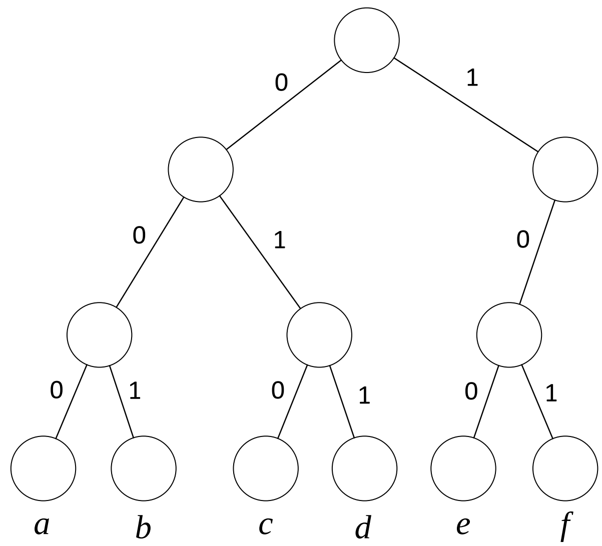
ניתן לייצג כל קוד חסר רישא ע"י עץ בינארי, כל תו מיוצג ע"י עלה.

מילת הקוד של התו מתוארת ע"י המסלול מהשורש לעלה, כאשר פנייה שמאלה תחשב כ- '0' ופנייה ימינה כ- '1'.

דוגמאות:

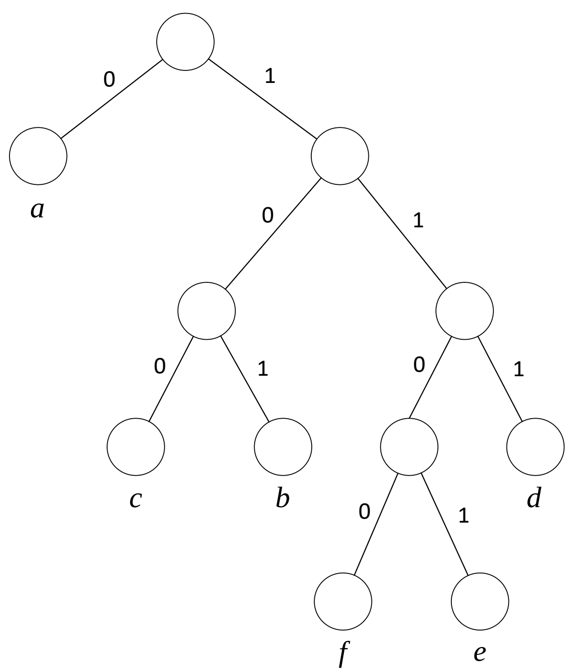
1) עץ אפשרות ראשונה

a	b	c	d	e	f
000	001	010	011	100	101



2) עץ אפשרות שנייה

a	b	c	d	e	f
0	101	100	111	1101	1100



קוד חסר רישא אופטימלי ייוצג ע"י עץ בינארי מלא. לכן מעכשיו נסתכל רק על קודים המיוצגים ע"י עץ בינארי מלא. כלומר אם העץ מייצג א"ב - C, אזי יש בו $|C|$ עלים ו- $|C|-1$ קודקודים פנימיים.

סימונים

עבור הא"ב C וקידוד שמיוצג ע"י העץ T, לכל תו $a \in C$:

- נסמן ב- $f(a)$ את מספר המופעים של a בקובץ. (תדירות - f)
- נסמן ב- $d_T(a)$ את העומק (אורך מילת הקוד) שמייצג את a ב- T.
- נסמן ב- $B(T)$ את אורך הקובץ המקודד ע"י T.

$$B(T) = \sum_{a \in C} f(a) \cdot d_T(a)$$

קוד האפמן

אלגוריתם חמדני לפתרון הבעיה שהומצא ע"י האפמן.

הבחירה החמדנית תהיה שהתווים בעלי מספר מופעים נמוך בקובץ יקודדו ע"י יותר ביטים.

המיון של התוים הוא לפי תדירות-כדי לעשות זאת משתמשים בערימת מינימום Q

בניית הפתרון מורכבת: בכל שלב ניקח את שני התוים בעלי התדירות הנמוכה ביותר , נהפוך אותם לאחים בעץ. לאבא שלהם ניתן את סכום התדירויות שלהם ונהפוך אותו ל"תו" חדש בעץ.

```
Huffman(C)
{
    n <- |C|
    Q <- C
    for i <- 1 .... n-1
    {
        create new node z
        x <- Extract_min(Q)
        y <- Extract_min(Q)
        z.left <- x
        z.right <- y
        z.f <- x.f+y.f
        insert(Q,z)
    }
    return Extract_min(Q)
}
```

סיבוכיות

בניית ערימה $O(n)$

יש n-1 איטרציות בכל איטרציה שלוש פעולות של ערימה בסיבוכיות $O(\log n)$

והשאר בסיבוכיות קבועה

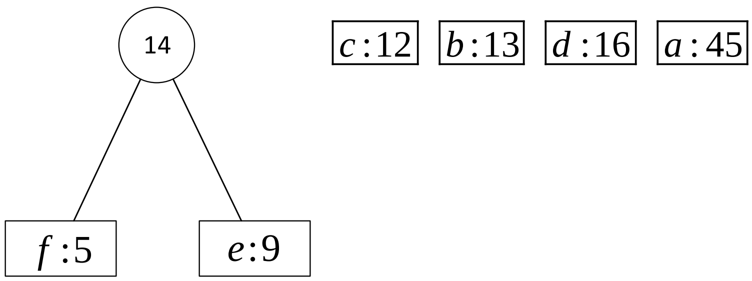
לכן סה"כ $O(n\log n)$

דוגמא לאלגוריתם:

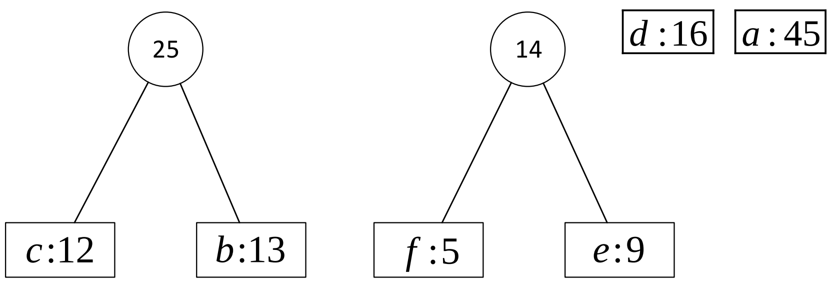
0) $n=6$ $\boxed{f:5}$ $\boxed{e:9}$ $\boxed{c:12}$ $\boxed{b:13}$ $\boxed{d:16}$ $\boxed{a:45}$

---

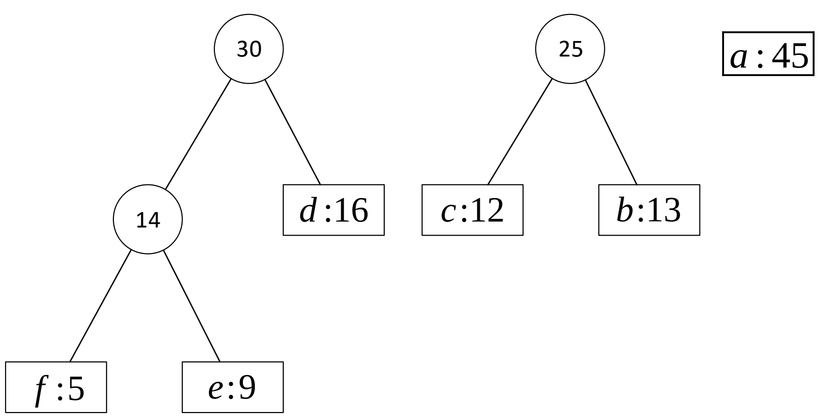
1)



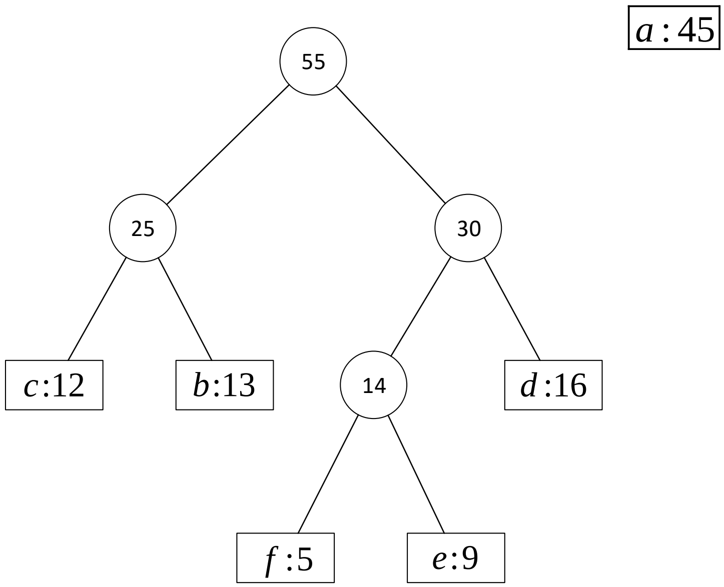
2)



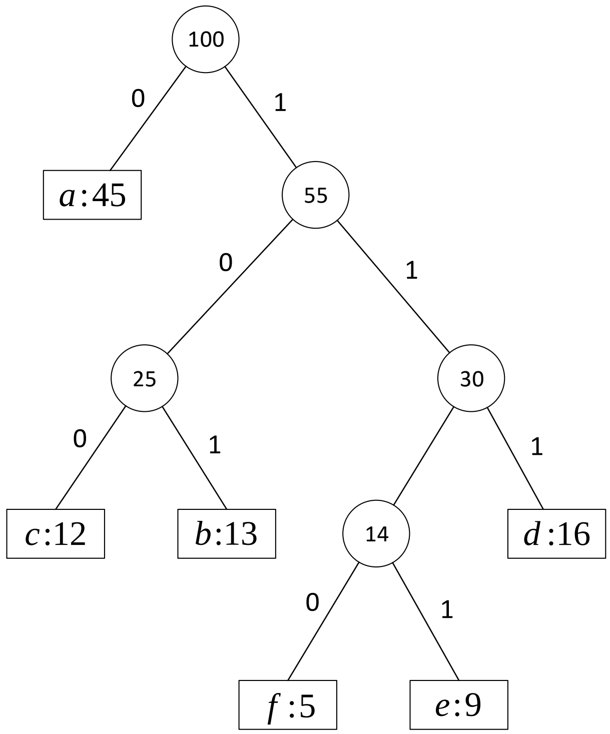
3)



4)



5)



אופטימליות של האלגוריתם

טענה

יהי C א"ב, ויהיו x ו- y התווים ב- C בעלי תדירות מינימאלית (כלומר $f(x)$ ו- $f(y)$ מינימליים). אזי קיים קוד חסר רישא אופטימלי שבעץ המייצג אותו x ו- y אחים בעומק המקסימלי בעץ, ושונים רק בתו האחרון במילת הקוד שלהם.

בלי הוכחה.

משפט

יהי C א"ב בעל n תווים. קוד האפמן מחזיר קוד אופטימלי עבור C.

הוכחה

נוכיח באינדוקציה על n.

בסיס האינדוקציה - $n=2$ ברור שהאפמן מחזיר קוד אופטימלי.

הנחת האינדוקציה - נניח שקוד האפמן על א"ב בעל n-1 תווים הוא אופטימלי.

צעד האינדוקציה - נוכיח עבור n.

יהי C א"ב בעל n תווים, נסמן ב- T_c את העץ המתקבל מאלגוריתם האפמן על C. נסמן ב- x ו- y את התווים בעלי התדירות המינימלית.

נבנה א"ב חדש C' כך: $C'=C\setminus\{x,y\}\cup\{z\}$ כך שלכל:

$$f_{C'}(z) = f_C(x) + f_C(y) \text{ ו- } f_C(a)=f_{C'}(a) \quad ,a\in C \leftarrow a\neq x\neq y$$

נסמן ב- $T_{C'}$ את העץ שמתקבל מקוד האפמן על C'.

נשים לב:

1) מהנחת האינדוקציה ומכך ש $|C'| = n-1$, $T_{C'}$ אופטימלי.

2) מפעולת האלגוריתם העץ T_C זהה ל- $T_{C'}$ חוץ מכך שב- T_C z קודקוד פנימי עם שני בנים עלים x ו- y.

כלומר לכל $a \neq x,y \in c$ $\quad d_{T_c}(a) = d_{T_{c'}}(a)$

$$d_{T_C}(x)=d_{T_C}(y)=d_{T_{C'}}(z)+1$$

$$f(x)d_{T_C}(x)+f(y)d_{T_C}(y)=\left(f(x)+f(y)\right)\left(d_{T_{C'}}(z)+1\right)=$$
$$=f(z)d_{T_{C'}}(z)+\left(f(x)+f(y)\right)$$

$$B(T_C)=\sum_{t\in C}f(t)d_{T_C}(t)=\sum_{t\neq z\in C'}f(t)d_{T_{C'}}(t)+f(x)d_{T_C}(x)+f(y)d_{T_C}(y)=$$
$$=\sum_{t\neq z\in C'}\left(f(t)d_{T_C}(t)\right)+f(z)d_{T_{C'}}(z)+f(x)+f(y)$$

$$= \mathrm{B}(\mathrm{T}_{c'}) + f(x) + f(y)$$
$$B(T_c) = B(T_{c'}) + f(x) + f(y)$$
$$B(T_{c'}) = B(T_c) - f(x) - f(y)$$

נניח בשלילה ש- T_C לא אופטימלי, כלומר קיים עץ T' שמתאים לא"ב של C כך ש- $B(T')<B(T_C)$

מהטענה ניתן להניח ש- x ו- y אחים בעלי עומק מקסימלי ב-T'

נבנה עץ חדש T'' על ידי השמטת x ו- y מT' והשארת האב שלהם z, בתור עלה.

$$f(z) = f(x) + f(y)$$

נשים לב ש- T'' קידוד עבור C'.

$$B(T'') = B(T') - f(x) - f(y) < B(T_C) - f(x) - f(y) = B(T_{C'})$$

$$\mathrm{B}(T'') < B(T_{C'})$$

בסתירה לכך ש- $T_{C'}$ אופטימלי ולכן T_C אופטימלי, כנדרש.