גירסה 1.00 - 31.1.2003

קודים פרפיקסים, עצי מצבים וקוד האפמן

מסמך זה הורד מהאתר http://underwar.livedns.co.il.
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות ל**ניר אדר**

Nir Adar
Email: underwar@hotmail.com
Home Page: http://underwar.livedns.co.il

אנא שלחו תיקונים והערות אל המחבר.

קודים פרפיסקים ועצי מצבים

הגדרות

נתון **א"ב** $\Sigma = \{0,1,2,...,\sigma-1\}$ א"ב בן σ אותיות.
מילה (מחרוזת אותיות) היא $c = a_1 a_2 ... a_l$, כך שמתקיים כי $\forall i, a_i \in \Sigma$.
l זהו אורך המילה.
קוד - $C = \{c_1, c_2, ..., c_m\}$ - אוסף של מילים, כאשר כל אחת מהמילים היא מילה מעל השפה.
הודעה - זוהי סדרת מילים משורשרת. קוד ייקרא **חד פענח** אם כל הודעה ניתנת לפענוח יחיד.
אם מחרוזת 1 מופיעה בתחילת מחרוזת שניה - אומרים שהמחרוזת הראשונה היא **רישא (פרפיקס)** של השניה.
אם מחרוזת 1 מופיעה בסוף מחרוזת שניה - אומרים שהמחרוזת הראשונה היא **סיפא (סופיקס)** של השניה.
קוד נקרא **קוד פרפיסקי** אם אין בו שתי מילות קוד שאחת מהן היא פרפיקס של השניה.

טענה

קוד פרפיקסי הוא קוד חד פענח.

הגדרה

עץ σ–מצבי הוא עץ מכוון שלכל צומת בו יש פוטנציאל ל- σ קשתות יוצאות הממוספרות $\Sigma = \{0,...,\sigma-1\}$. נגדיר בכינוי $w(x)$ את סימני המסלול מהשורש אל צומת x.

טענה א'

אם x אב של y, אזי $w(x)$ רישא של $w(y)$.

טענה ב'

אם $x \neq y$ אז $w(x) \neq w(y)$.

טענה ג'

אם $w(x)$ רישא של $w(y)$, נובע כי $w(x)$ רישא של $w(y)$.

טענה ד'

לכל קוד פרפיסקסי מעל א"ב עם σ אותיות, קיים עץ σ–מצבי.

הגדרה

עץ σ–מצבי מלא הוא עץ σ–מצבי שלכל צומת פנימית דרגה σ.

הגדרה

סכום אופייני של עץ σ–מצבי:

יהי $l_1, l_2, \ldots$ אורכי המסלולים לעלי העץ T. סכום אורכי העץ: $S(T) = \sum \sigma^{-l_i}$

טענה

בעץ מלא מתקיים: $S(T) = 1$.

הגדרה

סכום אופייני של קוד:

יהי $C = \{c_1, \ldots, c_n\}$ קוד. נגדיר: $S(c) = \sum \sigma^{-|c_i|}$

תוצאה

אם נתון קוד פרפיקסי, אז בהכרח הסכום האופייני קטן או שווה מ-1.

Huffman Code

מבוא

קוד זה עוסק בבעית דחיסת מידע.
נתון טקסט (באנגלית), כלומר - השפה היא בת 26 אותיות.
כיצד אנו מייצגים את השפה במשב? אנו יכולים לכתוב עבור כל אות מילה בינארית בת 5 ביטים, ולשרשר מילים כאלו על מנת להרכיב מילים ומשפטים.
בעייתיות בפתרון כזה: בקלט בעל n אותיות, גודל המסמך יהיה $5n$ ביטים.
בשנות ה-40 הציג שנון רעיון: אותיות שכיחות יותר יופיעו בצורה מקוצרת, ואותיות נדירות יותר יקבלו ייצוג ארוך יותר.

קוד Prefix

הקוד שנציג הוא קוד Prefix - אף מילה איננה התחלה של מילה אחרת.
לדוגמא: נניח שבידינו השפה $\Sigma = \{A, B, C, D\}$. נציג בחירה של קידוד לאותיות (קוד Prefix), בהתאם לרוח הרעיון של שנון.

אות	הסתברות	קידוד
A	0.4	0
B	0.3	10
C	0.2	110
D	0.1	111

נגדיר **הודעה** כהדבקה של מילים מהקוד שבחרנו. כל קוד שנבחר צריך לקיים כי ניתן לתרגם הודעה שנכתבת בו באופן יחיד למילה.
חשיבות קוד ה-Prefix: קוד זה חד-פענח באופן טריוויאלי, והוא נוח לפיענוח בזמן אמת.

אבחנה

האורך הממוצע של מילה בקוד הינו: $\overline{l} = p_1 l_1 + p_2 l_2 + ... + p_n l_n$

שאלה

השאלה אותה קוד Huffman פותר: איך בונים עבור ווקטור הסתברות נתון $(p_1, p_2, ..., p_n)$ את קוד ה-prefix שעבורו $\overline{l}$ הוא מינימום?

<u>פתרון</u>

נציג את הפתרון באמצעות דוגמא.

יהי ווקטור הסתברות כלשהו שלאחר מיון לפי גודל נקבל את ווקטור ההסתברות הבא:

0.6 0.2 0.1 0.05 0.03 0.01 0.01

נבצע את התהליך הבא:
בכל שלב, ניקח את שתי ההסתברויות הקטנות ביותר ונחבר אותן להסתברות אחת. נוציא את שתי ההסתברות מהווקטור, ונכניס במקומן בצורה ממויינת את ההסתברות החדשה.

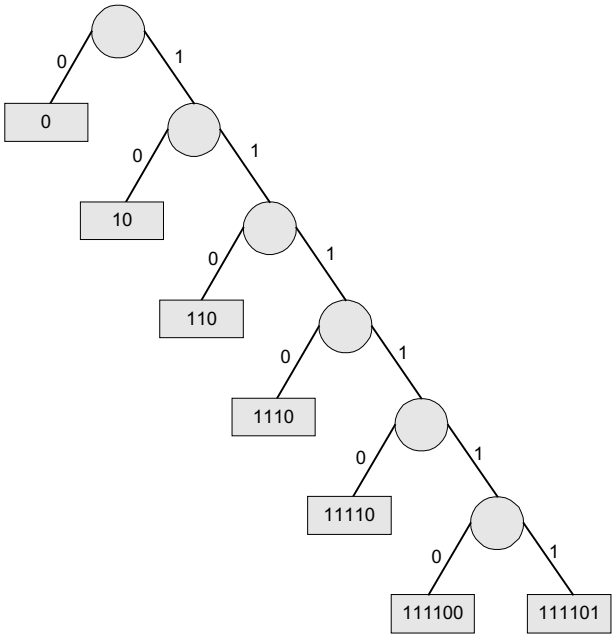
0.6 0.2 0.1 0.05 0.03 0.01 0.01
0.6 0.2 0.1 0.05 0.03 0.02
0.6 0.2 0.1 0.05 0.05
0.6 0.2 0.1 0.1
0.6 0.2 0.2
0.6 0.4

כעת נבוא לקדד את המילים:
ההסתברות 0.6 תקבל קידוד 0, וההסתברות 0.4 תקבל קידוד 1, ונמשיך בצורה הבאה:

<table>
<tr><td>0.6
0</td><td>0.4
1</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>0.2
10</td><td>0.2
11</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>0.1
110</td><td>0.1
111</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>0.05
1110</td><td>0.05
1111</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td>0.03
11110</td><td>0.02
11111</td></tr>
</table>

המספרים המודגשים הם הקידוד שאנו נותנים לכל מילה. ככל שמילה בהסתברות נמוכה יותר, היא תהיה ארוכה יותר.

נציג את המילים בעץ. נשתמש בקוד רק בצמתים המתאימים לעלים:



<u>הבחנה 1</u>

עץ אופטימלי הוא תמיד מלא.

<u>הבחנה 2</u>

אם $p_i > p_j$, אז בכל עץ אופטימלי יתקיים כי $p_i \geq p_j$, כלומר יושב גבוה יותר בעץ או באותה רמה.

<u>הבחנה 3</u>

עבור כל ווקטור הסתברות ממוין נתון $(p_1, p_2, ..., p_n)$, קיים עץ אופטימלי שבו מתקיים $p_1 \geq p_2 \geq ...$, כך ש-p_n, p_{n-1} הם ההסתברויות הקטנות ביותר.
p_n, p_{n-1} הם עלים אחים.