Huffman

CS4800

abhi shelat
MOSCOW — President Vladimir V. Putin’s typically theatrical order to withdraw the bulk of Russian forces from Syria, a process that the Defense Ministry said it began on Tuesday, seemingly caught Washington, Damascus and everybody in between off guard — just the way the Russian leader likes it.

By all accounts, Mr. Putin delights at creating surprises, reinforcing Russia’s newfound image as a sovereign, global heavyweight and keeping him at the center of world events.
MOSCOW — President Vladimir V. Putin’s typically theatrical order to withdraw the bulk of Russian forces from Syria, a process that the Defense Ministry said it began on Tuesday, seemingly caught Washington, Damascus and everybody in between off guard — just the way the Russian leader likes it.

By all accounts, Mr. Putin delights at creating surprises, reinforcing Russia’s newfound image as a sovereign, global heavyweight and keeping him at the center of world events.
\( c \in C \quad f_c \quad T \)

\[
\begin{array}{ll}
\varepsilon & 235 \\
i & 200 \\
o & 170 \\
u & 87 \\
p & 78 \\
g & 47 \\
b & 40 \\
f & 24 \\
\end{array}
\]

881
<table>
<thead>
<tr>
<th>$c \in C$</th>
<th>$f_c$</th>
<th>$T$</th>
<th>$\ell_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>e: 235</td>
<td><strong>000</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>i: 200</td>
<td><strong>001</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>o: 170</td>
<td><strong>010</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>u: 87</td>
<td><strong>011</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>p: 78</td>
<td><strong>100</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>g: 47</td>
<td><strong>101</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>b: 40</td>
<td><strong>110</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>f: 24</td>
<td><strong>111</strong></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>881</td>
<td></td>
<td>$\rightarrow$</td>
<td><strong>2643</strong></td>
</tr>
</tbody>
</table>
def: cost of an encoding

\[ B(T, \{f_c\}) = \sum_{c \in C} f_c \cdot l_c \]

code frequency of your messages

length of code for character \( c \) in code \( T \)

<table>
<thead>
<tr>
<th>( c \in C )</th>
<th>( f_c )</th>
<th>( T )</th>
<th>( l_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>235</td>
<td>000</td>
<td>3</td>
</tr>
<tr>
<td>i</td>
<td>200</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>o</td>
<td>170</td>
<td>010</td>
<td>3</td>
</tr>
<tr>
<td>u</td>
<td>87</td>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>p</td>
<td>78</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>g</td>
<td>47</td>
<td>101</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>40</td>
<td>110</td>
<td>3</td>
</tr>
<tr>
<td>f</td>
<td>24</td>
<td>111</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ 881 \cdot 3 = 2643 \]
character frequency

e: 234803
i: 200613
a: 198938
o: 170392
r: 160491
n: 158281
t: 152570
s: 139238
l: 130172
c: 103307
u: 78211
p: 78077
m: 70504
d: 68007
h: 64165
y: 51527
g: 47011
b: 40351
f: 2110
v: 20103
k: 16912
w: 13825
z: 8439
x: 6926
q: 3729
j: 3075
International Morse Code

- 1 dash = 3 dots
- The space between parts of the same letter = 1 dot.
- The space between letters = 3 dots.
- The space between words = 7 dots.

A: •••  V: •••
B: ••  W: •••••
C: ••••  X: ••••
D: •••••  Y: •••••
E: •  Z: ••••
F: •••••  .: •••••
G: •••••  ,: •••••
H: •••••••  7: •••••
I: ••••••  /: •••
J: •••••••  @: •••
K: ••••••  1: •••••
L: ••••••  2: •••••
M: ••••••  3: •••••
N: •••••••  4: •••••
O: •••••••  5: •••••
P: •••••••  6: •••••
Q: •••••••  7: •••••
R: •••••••  8: •••••
S: ••••••  9: •••••
T: •••••••  0: •••••
U: •••••••  ••••••
International Morse Code

- 1 dash = 3 dots
- The space between parts of the same letter = 1 dot
- The space between letters = 3 dots
- The space between words = 7 dots

A: ---
B: --
C: ----
D: --
E: .
F: ..
G: ---
H: ---
I: .
J: ---
K: -
L: .---
M: --
N: ---
O: ---
P: .--
Q: --.-
R: .-.
S: ---
T: -
U: ---
V: .--.
W: .--
X: -..-
Y: -.--
Z: --.

A: ---
E: .
T: -
E: .
T: -
R: T
def: prefix-free code

Code such that for any two symbols $x, y \in C, x \neq y \implies \text{code}(x) \text{ is not a prefix of code}(y)$
def: prefix-free code

\forall x, y \in C, x \neq y \implies \text{CODE}(x) \text{ not a prefix of } \text{CODE}(y)
def: prefix code

∀x, y ∈ C, x ≠ y → CODE(x) not a prefix of CODE(y)

e: 235
i: 200
o: 170
u: 87
p: 78
f: 24

g: 47
b: 40

□
□
□
□
□
□
□
□
decoding a prefix code

e: 235  0
i: 200  10
o: 170  110
u: 87   1110
p: 78   11110
g: 47   111110
b: 40   1111110
f: 24   11111110
code to binary tree

e: 235  
0
i: 200  10
o: 170  110
u: 87  1110
p: 78  11110

111110

1111110

11111110

111111010111110
prefix code

binary tree
use tree to encode

<table>
<thead>
<tr>
<th>c ∈ C</th>
<th>fc</th>
<th>T</th>
<th>ℓc</th>
</tr>
</thead>
<tbody>
<tr>
<td>e:</td>
<td>235</td>
<td>00</td>
<td>2</td>
</tr>
<tr>
<td>i:</td>
<td>200</td>
<td>01</td>
<td>2</td>
</tr>
<tr>
<td>o:</td>
<td>170</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>u:</td>
<td>87</td>
<td>110</td>
<td>3</td>
</tr>
<tr>
<td>p:</td>
<td>78</td>
<td>111</td>
<td>3</td>
</tr>
</tbody>
</table>
given the frequencies for some message space $\mathcal{M}$,

design an optimal prefix-free code.
goal

(All frequencies are > 0)

given the character frequencies \( \{ f_c \} \), \( c \in C \)

produce a prefix code \( T \) with smallest cost

\[
\min_T B(T, \{ f_c \})
\]
lemma: optimal tree must be full.

Proof: Suppose a code had a node with only 1 child. One could remove this node to produce a code with shorter codewords for several symbols.
divide & conquer?
counter-example

e: 32
i: 25
o: 20
u: 18
p: 5

(18.2 + 20.3 = 96
20.2 + 18.3 = 94

e: 235 01
i: 200 11
o: 170 10
u: 87 0011
p: 78 0010
g: 47 0000
b: 40 00011
f: 24 00010
objective

Prove that the Huffman algorithm produces the optimal prefix-free code.

In search of an exchange argument.
Exchange argument

Lemma: Let $f_x$ and $f_y$ be the 2 smallest frequencies in $E_{fc \beta}$. There exists an optimal code such that $x$ and $y$ are siblings.
lemma:

Let \( x, y \in C \) be characters with smallest frequencies \( f_x, f_y \). There exists an optimal prefix code \( T'' \) for \( C \) in which \( x, y \) are siblings. That is, the codes for \( x, y \) have the same length and only differ in the last bit.

Proof: Let \( T \) be an optimal code for \( \mathbb{Z}_{\leq 3} \).

- If \( x \) and \( y \) are siblings in \( T \), the claim holds.
- Otherwise, let \( a \) and \( b \) be the 2 symbols at greatest depth in \( T \) (which are siblings).

Why do \( a \) & \( b \) exist??

By \( T \) is optimal, \( A \) as we argued before, nodes only have 0 or 2 children in optimal codes.
Let $x, y \in C$ be characters with smallest frequencies $f_x, f_y$. There exists an optimal prefix code $T''$ for $C$ in which $x, y$ are siblings. That is, the codes for $x, y$ have the same length and only differ in the last bit.
Let $x, y \in C$ be characters with smallest frequencies $f_x, f_y$. There exists an optimal prefix code $T''$ for $C$ in which $x, y$ are siblings. That is, the codes for $x, y$ have the same length and only differ in the last bit.
exchange argument

Exchange a with $x$ in the tree to construct a new tree $T'$. 

first step
exchange argument

\[ B(T) = z + f_x \cdot l_x + f_a \cdot l_a \]

\[ B(T') = z + f_a \cdot l_x + f_x \cdot l_a \]
\[ \text{B(C)} = x + fx \cdot lx + fa \cdot la \]

\[ \text{B(C')} = x + fa \cdot lx + fx \cdot la \]

\[ \text{B(C)} - \text{B(C')} = fx(lx - la) - fa(lx - la) \geq 0 \]

\[ = (fx - fa)(lx - la) \leq 0 \]

\[ \text{But } T \text{ is optimal, and so } \text{B(C)} = \text{B(C')} \]
$B(T) = \sum_c f_c \ell_c + f_x \ell_x + f_a \ell_a$

$B(T') = \sum_c f'_c \ell'_c + f'_x \ell'_x + f'_a \ell'_a$

$B(T) - B(T') \geq 0$
exchange argument

\[
B(T') - B(T'') \geq 0
\]
\[ B(T) - B(T') \geq 0 \quad \text{and} \quad B(T') - B(T'') \geq 0 \]

\[ B(T) - B(T'') \geq 0 \]

But again, b/c \( T \) is optimal, they must be equal, \( \Rightarrow T'' \) is optimal.
$B(T) - B(T') \geq 0$  

$B(T') - B(T'') \geq 0$

$T''$ is also optimal
Lemma: Let $x, y \in C$ be characters with smallest frequencies $f_x, f_y$. There exists an optimal prefix code $T''$ for $C$ in which $x, y$ are siblings. That is, the codes for $x, y$ have the same length and only differ in the last bit.
optimal sub-structure

\[ f_c \begin{array}{cccccc} 235 & 200 & 170 & 87 & 78 & 47 \end{array} \begin{array}{cc} f_x & f_y \end{array} \]

problem of size \( n \)
optimal sub-structure

\[
\begin{align*}
&f_c \quad 235 \quad 200 \quad 170 \quad 87 \quad 78 \quad 47 \quad 40 \quad 24 \\
&\text{problem of size } n
\end{align*}
\]

\[
\begin{align*}
&f_{c'} \quad 235 \quad 200 \quad 170 \quad 87 \quad 78 \quad 47 \quad 64 \\
&\text{problem of size } n-1
\end{align*}
\]

optimal solution to \( f_{c'} \), and then replace \( f_z \) with \( \varnothing \)

the resulting tree is optimal for \( f_c \).
optimal sub-structure

\[ f_c \begin{bmatrix} 235 & 200 & 170 & 87 & 78 & 47 & 40 & 24 \end{bmatrix} \]

problem of size \( n \)

\[ f_{c'} \begin{bmatrix} 235 & 200 & 170 & 87 & 78 & 47 & 64 \end{bmatrix} \]

problem of size \( n-1 \)

Lemma:
optimal sub-structure

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>235</th>
<th>200</th>
<th>170</th>
<th>87</th>
<th>78</th>
<th>47</th>
<th>$f_x$</th>
<th>$f_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{c'}$</td>
<td>235</td>
<td>200</td>
<td>170</td>
<td>87</td>
<td>78</td>
<td>47</td>
<td>64</td>
<td>$f_z$</td>
</tr>
</tbody>
</table>

problem of size $n-1$
Lemma: The optimal solution for \( T \) consists of computing an optimal solution for \( T' \) and replacing the left \( z \) with a node having children \( x, y \).
\[
B(T) = B(T') - \int_{z} l_{z} + (l_{z} + 1) (f_{x} + f_{y}) \\
= B(T') + f_{x} + f_{y}
\]
\[ B(T') = B(T) - f_x - f_y \]
Suppose $T$ is not optimal.
Suppose $T$ is not optimal.

There is some other tree $U$ such that $B(U) < B(T)$. 
Suppose $T$ is not optimal.

\[
B(U) < B(T) = B(T') + f_x + f_y
\]

\[
B(U') < B(T')
\]

This suggests that $T'$ was not an optimal solution. This is a contradiction.
Suppose $T$ is not

$$B(U) < B(T')$$

$$B(U') = B(U) - f_x - f_y$$

$$< B(t) - f_x - f_y$$

But this implies that $B(T')$ was not optir
therefore