Data Structures and Algorithms – (COMP SCI 2C03)
Winter 2021
Tutorial - 9

April 5, 2021

1. How many letter comparisons would KMP algorithm perform on the text $a^n$ and pattern $a^{m-1}b$.
   **Answer:** For the text $a^n$ and pattern $a^{m-1}b$, KMP would perform $2n - m$ letter comparisons.

2. Compute the border array of the string `abaababaabaab`
   **Answer:**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | a | b | a | a | b | a | b | a | a | b | a | a | b |
| $\beta_w$ | 0 | 0 | 1 | 1 | 2 | 3 | 2 | 3 | 4 | 5 | 6 | 4 | 5 |

3. Compute the border array of the string $w = a^n$.
   **Answer:** $\beta_w = 0123 \ldots n - 1$

4. Consider the four variable-length codes shown in Figure 1. Which of the codes are prefix-free? Which of the codes are uniquely decodable? For those that are uniquely decodable, give the encoding of `AABCABABACAABDAA`.
   **Answer:** Code 1 **is not** prefix-free as 10 is prefix of 100.
   Code 2 **is not** prefix-free as 0 is prefix of 00 and 1 is prefix of 11.
   Code 3 **is** uniquely decodable:
   $\overset{A}{1}\ \overset{A}{1}\ \overset{B}{01}\ \overset{C}{001}\ \overset{A}{1}\ \overset{B}{01}\ \overset{A}{1}\ \overset{B}{01}\ \overset{A}{1}\ \overset{C}{001}\ \overset{A}{1}\ \overset{A}{1}\ \overset{B}{01}\ \overset{D}{0001}\ \overset{A}{1}\ \overset{A}{1}$

Code 4 **is** uniquely decodable:

$\overset{A}{1}\ \overset{A}{1}\ \overset{B}{01}\ \overset{C}{001}\ \overset{A}{1}\ \overset{B}{01}\ \overset{A}{1}\ \overset{B}{01}\ \overset{A}{1}\ \overset{C}{001}\ \overset{A}{1}\ \overset{A}{1}\ \overset{B}{01}\ \overset{D}{000}\ \overset{A}{1}\ \overset{A}{1}$

| symbol | code 1 | code 2 | code 3 | code 4 |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 100 | 1 | 01 | 01 |
| C | 10 | 00 | 001 | 001 |
| D | 11 | 11 | 0001 | 000 |

Figure 1: Table for Question 4

5. How many bits are needed to encode N copies of the symbol a, as a function of N) using run-length encoding? How many bits are needed to encode N copies of abc (as a function of N) using run-length encoding? (you may consider ASCII encoding)
   **Answer:** If we use ASCII encoding and run-length encoding for alternating runs of zeros and ones, then we need $32N$ bits to encode N copies of the symbol a (see next question for more details), and $96N$ bits to encode N copies of abc.

6. Give the result of encoding the strings a, aa, aaa, aaaa, ... (strings consisting of N a's) with run-length and Huffman encoding.
   **Answer:** 8-bit ASCII code for **a**: 01100001.
   String "a":

   - run-length: $\overset{\#0}{00000001}\ \overset{\#1}{00000010}\ \overset{\#0}{00000100}\ \overset{\#1}{00000001}$
   - Huffman: $\overset{leaf}{1}\ \ \overset{ASCII(a)}{01100001}\ \overset{a}{0}$

   String "aa":

   - run-length: $\overset{\#0}{00000001}\ \ \overset{\#1}{00000010}\ \ \overset{\#0}{00000100}\ \ \overset{\#1}{00000001}\ \ \overset{\#0}{00000001}$ $\overset{\#1}{00000010}\ \ \overset{\#0}{00000100}\ \ \overset{\#1}{00000001}$
   - Huffman: $\overset{leaf}{1}\ \ \overset{ASCII(a)}{01100001}\ \overset{a}{0}\ \overset{a}{0}$

String “N a’s”:

- run-length: $\overset{\#0}{00000001}\ \overset{\#1}{00000010}\ \underset{\times N}{}\ \overset{\#0}{00000100}\ \overset{\#1}{00000001}$
- Huffman: $\overset{leaf}{1}\ \ \overset{ASCII(a)}{01100001}\ \ \underset{\times N}{\overset{a}{0}}$

7. Compute the bitstream encoding of the binary trie given in Figure 2 representing prefix-free codes for letters A, B, C, D, E.

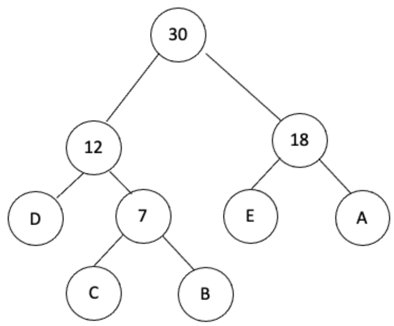


Figure 2: Prefix-free code binary trie

**Answer:** The preorder traversal of the trie is: 30 12 D 7 C B 18 E A. Marking internal nodes with bit 0, and character with bit 1 followed by its 8-bit ASCII code, we get the following bitstream encoding:
0 0 101000100 0 101000011 101000010 0 101100101 101000001.

8. Consider the input string $w = abababacaaaadaeaceeaabbb$. Encoding $w$ in 8-bit ASCII requires how many bits? How much savings in terms of bits is achieved if $w$ is encoded using Huffman encoding (remember to include the number of bits required to encode the binary trie)?
**Answer:** For 8-bit ASCII, we have 24 characters and each needs 8 bits, so we need in total of $24 \times 8 = 192$ bits.

The Huffman Trie is provided in Figure 3. This is the encoding of the Trie:

$\overset{\overset{1}{\downarrow}}{0}\ \ \overset{\overset{leaves}{\color{red}\downarrow}}{1}\ \ \overset{a}{01100001}\ \overset{\overset{2}{\downarrow}}{0}\ \overset{\color{red}\downarrow}{1}\ \overset{b}{01100010}\ \overset{\overset{3}{\downarrow}}{0}\ \overset{\color{red}\downarrow}{1}\ \overset{e}{01100101}\ \overset{\overset{4}{\downarrow}}{0}\ \overset{\color{red}\downarrow}{1}\ \overset{d}{01100100}\ \overset{\color{red}\downarrow}{1}\ \overset{c}{01100011}$

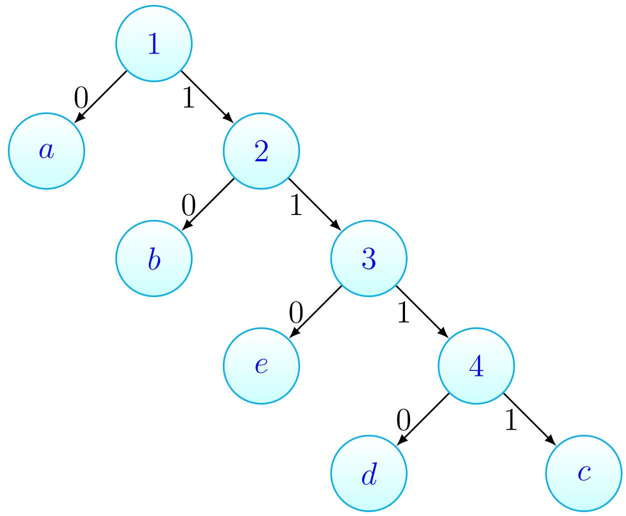

Figure 3: Huffman Trie for Q8

We need 49 bits to encode the trie. The result of encoding the string w needs 45 bits as follow:

$\overset{a}{0}\ \overset{b}{10}\ \overset{a}{0}\ \overset{b}{10}\ \overset{a}{0}\ \overset{b}{10}\ \overset{a}{0}\ \overset{c}{1111}\ \overset{a}{0}\ \overset{a}{0}\ \overset{a}{0}\ \overset{a}{0}\ \overset{d}{1110}\ \overset{a}{0}\ \overset{e}{110}\ \overset{a}{0}\ \overset{c}{1111}\ \overset{e}{110}\ \overset{e}{110}\ \overset{a}{0}\ \overset{a}{0}\ \overset{b}{10}\ \overset{b}{10}\ \overset{b}{10}$

In total, we need 94 bits with Huffman code. Compare to 8-bit ASCII, we have $\dfrac{94}{192} = 48\%$ compression ratio.