Huffman Coding

J.-S. Roger Jang (張智星)

MIR Lab, CSIE Dept.

National Taiwan University

jang@mirlab.org, http://mirlab.org/jang

2019/5/28

Optimal Merge Pattern

- Optimal merge pattern
  - Goal: An optimal way of merge n sorted lists, where the merge cost is proportional to the length of the lists to be merged.
  - Fact: greedy algorithm works
  - Application: Huffman coding

How to Merge Two Sorted Lists?

- A and B are two sorted lists:
  - A: 1 3 7 9 12 25
  - B: 2 5 10 11 13 15 17 27 39
- To merge A and B into C:
  - Use two pointer to point to the first elements
  - Output the small one and advance the pointer
- Complex of merge: O(|A|+|B|)=O(m+n)

How to Merge n Sorted Lists?

- N=3
  - Lists: A, B, C
  - Sizes: 1, 5, 2
- N=4
  - Lists: A, B, C, D
  - Sizes: 2, 5, 3, 8
- Greedy algorithm works!
  - Objection function = $\sum_{i=1}^{n} s_i d_i$

Example of Optimal Merge Pattern

- Optimal merge pattern
  - Lists: A, B, C, D, E
  - Sizes: 2, 5, 4, 3, 8

Quiz!

Huffman Coding

- Goal: To encode a message to be sent or stored with the least no. of bits ➔ lossless data compression
- Example
  - Message: bccabbddaeccbbaeddcc
  - Simple encoding: To encode each character as a ASCII code
    - a ➔ 97 = 01100001
    - b ➔ 98 = 01100010
    - c ➔ 99 = 01100011
    - d ➔ 100 = 01100100
    - e ➔ 101 = 01100110

Message Encoding by ASCII

- Example
  - Message: bccabbddaeccbbaeddcc
- Simple coding by ASCII
  - Simple encoding: To encode each character as a ASCII code
    - a → 97 = 01100001
    - b → 98 = 01100010
    - c → 99 = 01100011
    - d → 100 = 01100100
    - e → 101 = 01100110
  - Total bits = 8*20 = 160 bits

Message Encoding by Custom Table

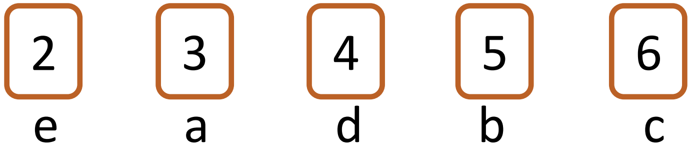
- Example
  - Message: bccabbddaeccbbaeddcc
- Fixed-length coding by a custom table
  - Encoding via a custom table, with 3 bits for each character
    - a → 000
    - b → 001
    - c → 010
    - d → 011
    - e → 100
  - Total bits = 3*20 + 8*5 + 3*5 = 115 bits
    - Message: 3*20 bits
    - Characters: 8*5 bits
    - Codes: 3*5 bits

Huffman Encoding

- Example
  - Message: bccabbddaeccbbaeddcc

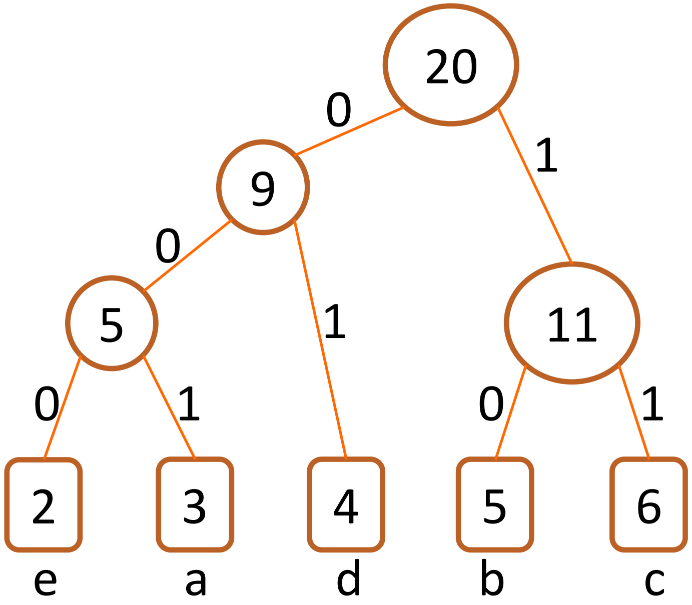
Quiz!

- Variable-length coding proposed by Huffman in 1951
  - Encoding via a custom table, with 3 bits for each character
    - a: count=3, code=001 ➔ 3*3 = 9 bits
    - b: count=5, code= 10 ➔ 2*5 = 10 bits
    - c: count=6, code= 11 ➔ 2*6 = 12 bits
    - d: count=4, code= 01 ➔ 2*4 = 8 bits
    - e: count=2, code=000 ➔ 3*2 = 6 bits
  - Total bits = 45 + 8*5 + 12 = 97 bits
    - Message: 45 bits
    - Characters: 8*5 bits
    - Codes: 3+2+2+2+3 = 12 bits



Decoding

- Original message: bccabbddaeccbbaeddcc
- Encoded: 10111100110100101001…
- Follow the tree to do decoding (just like tries)



b

Exercise

- Use Huffman coding to encode “catch the cat”.
  - What is the count for each character (including space)?
  - Draw the Huffman tree. What is the code for each character?
  - What is the total no. of bits for this coding scheme?

Youtube Tutorials

- Optimal merge pattern
- Huffman coding