COMP90038 Algorithms and Complexity

Huffman Encoding for Data Compression

Douglas Pires

Lecture 21

Semester 1, 2021

Data Compression

From an information-theoretic point of view, most computer files contain much redundancy.

Compression is used to store files in less space.

For text files, savings up to 50% are common.

For binary files, savings up to 90% are common.

Savings in space mean savings in time for file transmission.

Run-Length Encoding

For a text that has long runs of repeated characters, we could compress by counting the runs:

AAAABBBAABBBBBCCCCCCCCDABCBAAABBBBCCCD

can then be encoded as

4A3BAA5B8CDABCB3A4B3CD

For English text this is not very useful.

For binary files it can be very good.

Run-Length Encoding

```
000000000000000000000000000011111111111111000000000
000000000000000000000000001111111111111111110000000
000000000000000000000001111111111111111111111110000
000000000000000000000011111111111111111111111111000
000000000000000000001111111111111111111111111111110
000000000000000000011111100000000000000000001111111
000000000000000000011111000000000000000000000011111
000000000000000000011100000000000000000000000000111
000000000000000000011100000000000000000000000000111
000000000000000000011100000000000000000000000000111
000000000000000000011100000000000000000000000000111
000000000000000000001110000000000000000000000001110
000000000000000000000011100000000000000000000111000
011111111111111111111111111111111111111111111111111
011111111111111111111111111111111111111111111111111
011111111111111111111111111111111111111111111111111
011111111111111111111111111111111111111111111111111
011000000000000000000000000000000000000000000000011
```

Variable-Length Encoding

Instead of using the usual 8 bits for characters, assign characters codes in such a way that common characters have shorter codes.

For example, E could have code 0.

But then no other character code can start with 0.

In general, no character's code can be a prefix of some other character's code (unless we somehow put separators between characters).

Variable-Length Encoding

Suppose we count symbols and find these numbers of occurrences:

Symbol	Weight
B	4
D	5
G	10
F	12
C	14
E	27
A	28

Here are some sensible codes that we may use for symbols:

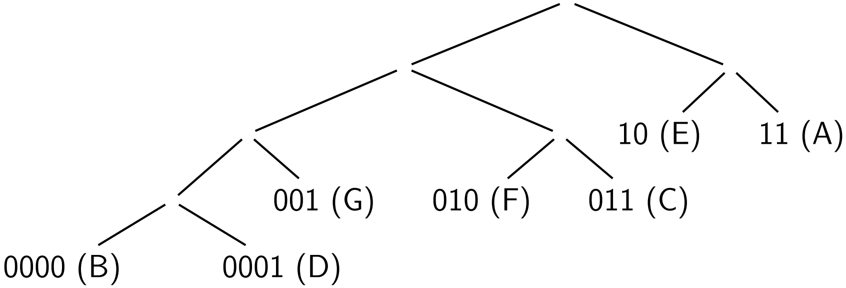
Symbol	Code
A	11
B	0000
C	011
D	0001
E	10
F	010
G	001

Tries for Variable-Length Encoding

A trie is a binary tree for search applications.

To search for a key we look at individual bits of a key and descend to the left whenever a bit is 0, to the right whenever it is 1.

Using a trie to determine codes means that no code will be the prefix of another!



Variable-Length Encoding

With the codes given by the trie, we can represent

FACE

in just 10 bits:

$$\underbrace{010}_{\mathrm{F}}\underbrace{11}_{\mathrm{A}}\underbrace{011}_{\mathrm{C}}\underbrace{10}_{\mathrm{E}}$$

If we were to assign 3 bits per character, FACE would take 12 bits.

Encoding a Message

We shall shortly see how to design the codes for symbols, taking symbol frequencies into account.

Once we have a table of codes, encoding is straightforward.

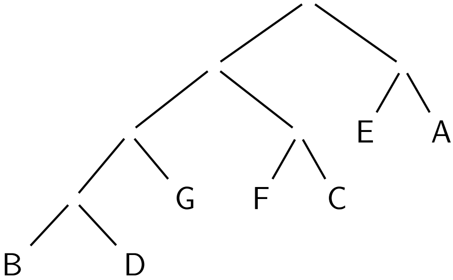
For example, to encode ‘BAGGED’, simply concatenate the codes for B, A, G, G, E and D:

000011001001100001

A	11
B	0000
C	011
D	0001
E	10
F	010
G	001

Decoding a Message

To decode 000011001001100001, use the trie, repeatedly starting from the root, and printing each symbol found as a leaf.



Huffman Encoding: Choosing the Codes

Sometimes (for example for common English text) we may know the frequencies of letters fairly well.

If we don't know about frequencies then we can still count all characters in the given text as a first step.

But how do we assign codes to the characters once we know their frequencies?

By repeatedly selecting the two smallest weights and fusing them.

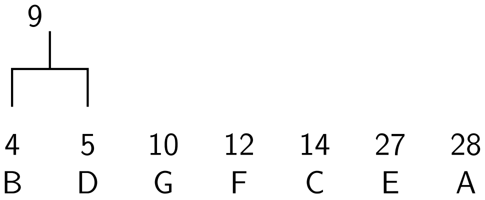
This is Huffman's algorithm—another example of a greedy method.

The resulting tree is a Huffman tree.

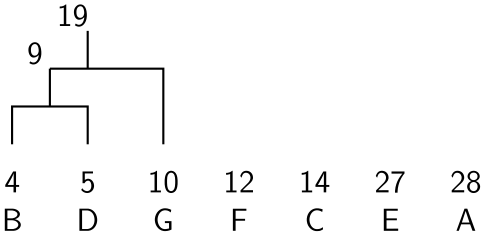
Huffman Trees

4	5	10	12	14	27	28
B	D	G	F	C	E	A

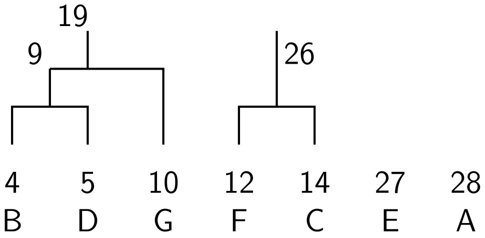
Huffman Trees



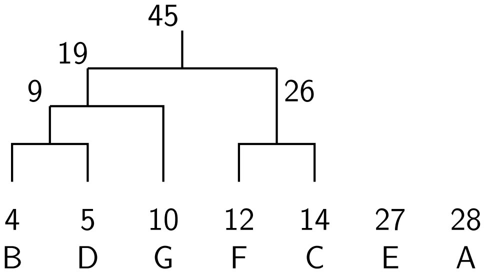
Huffman Trees



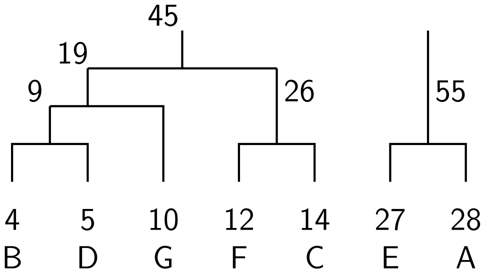
Huffman Trees



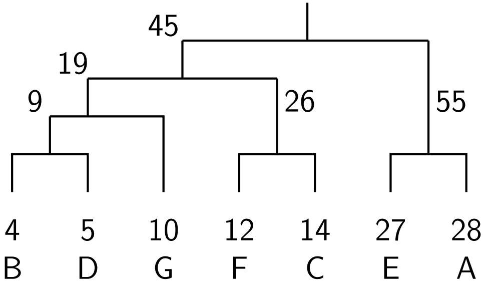
Huffman Trees



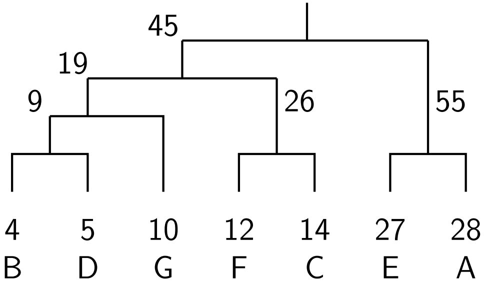
Huffman Trees



Huffman Trees



Huffman Trees



We end up with the trie from before!

One can show that this gives the best encoding.

Compressed Transmission

If the compressed file is being sent from one party to another, the parties must agree about the codes used.

Alternatively, the trie must be sent along with the message.

For long files this extra cost is negligible.

Modern variants of Huffman encoding, like Lempel-Ziv compression, assign codes not to individual symbols but to sequences of symbols.

The Final Week of This Semester

We will briefly discuss complexity theory (a large topic), NP-completeness, and approximation algorithms.

We will also briefly review the material covered in the subject.