Compression from scratch

Purdue Hackers

Kian Kasad

Grab these slides!

https://puhack.horse/compression-workshop

Before we start...

Join the Purdue Hackers Discord!

https://puhack.horse/discord

Open the Python notebook

https://puhack.horse/compression-py

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes to build Huffman encoding
- 30 minutes for you to explore!
  - Several options:
    - Implement Huffman decoding
    - Learn & implement Lempel-Ziv (LZ77)
    - Combine Huffman & LZ77 to get DEFLATE

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes on Huffman implementation
- 30 minutes for you to explore!
    - Several options:
        - Implement Huffman decoding
        - Learn & implement Lempel-Ziv (LZ77)
        - Combine Huffman & LZ77 to get DEFLATE

Run-length encoding (RLE)

A A A A A

Run-length encoding (RLE)

$$\text{AAAAA} \rightarrow \text{5A}$$

Run-length encoding (RLE)

Wooooooooooo! → W4oh6o!

Run-length encoding (RLE)

```
                             ^
                ________    ^^^
               |xxxxxxx|  _^^^^^_
               |xxxxxxx| | [][]  |
            ______xxxxx| |[][][] |
           |++++++|xxxx| | [][][]|      METROPOLIS
           |++++++|xxxx| |[][][] |
           |++++++|_________ [][]|
           |++++++|=|=|=|=|=| [] |
           |++++++|=|=|=|=|=|[][]|
___________|++HH++|  _HHHH__|   _________   _________  _________
         _______________   ______________      ______________
__________________  ___________    __________________    ____________
```

Questions?

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes on Huffman implementation
- 30 minutes for you to explore!
  - Several options:
    - Implement Huffman decoding
    - Learn & implement Lempel-Ziv (LZ77)
    - Combine Huffman & LZ77 to get DEFLATE

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes on Huffman implementation
- 30 minutes for you to explore!
  - Several options:
    - Implement Huffman decoding
    - Learn & implement Lempel-Ziv (LZ77)
    - Combine Huffman & LZ77 to get DEFLATE

Huffman coding

- Plain ASCII text: each character = 1 byte = 8 bits
- Some characters are common; some aren’t
- What if we use fewer bits for common characters?

Huffman coding

- Main idea:

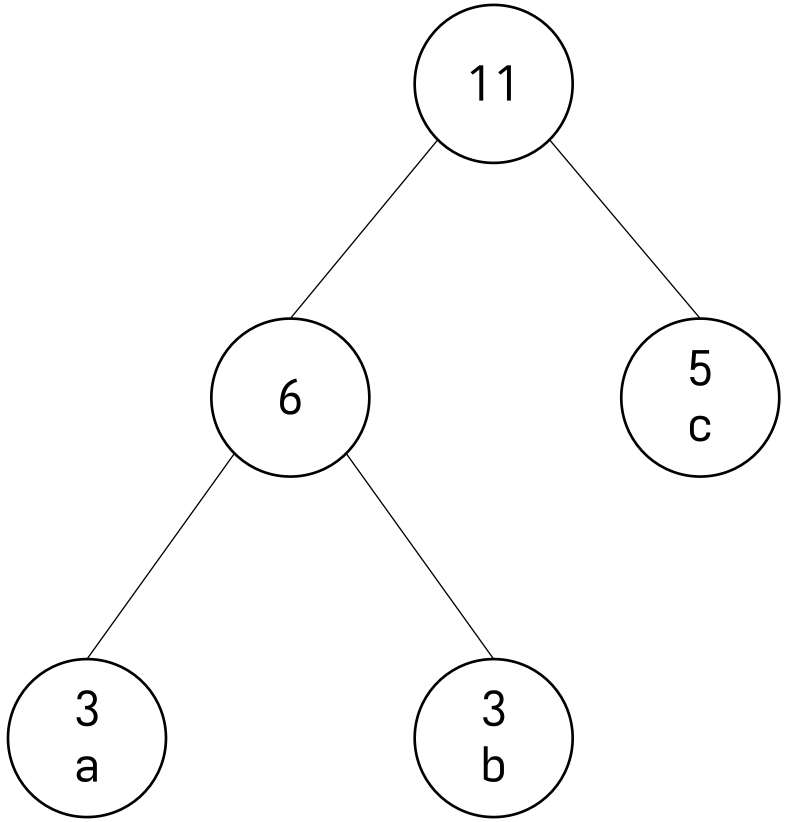
    Frequent characters get fewer bits.

    Infrequent characters get more bits.

How to do this?

Build a tree!

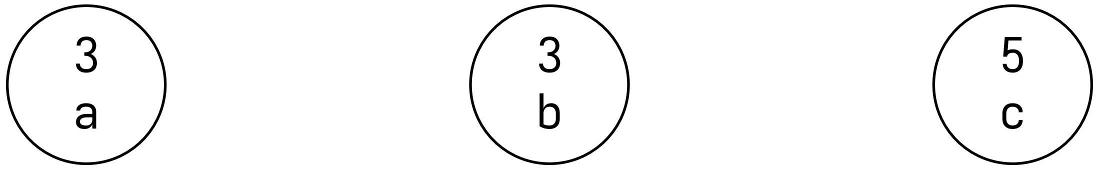
aaabbbccccc



Questions?

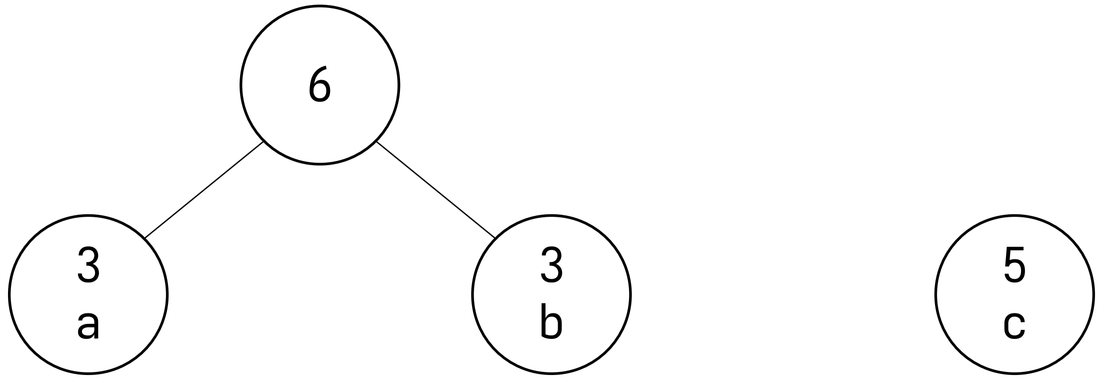
How to do this?

aaabbbccccc



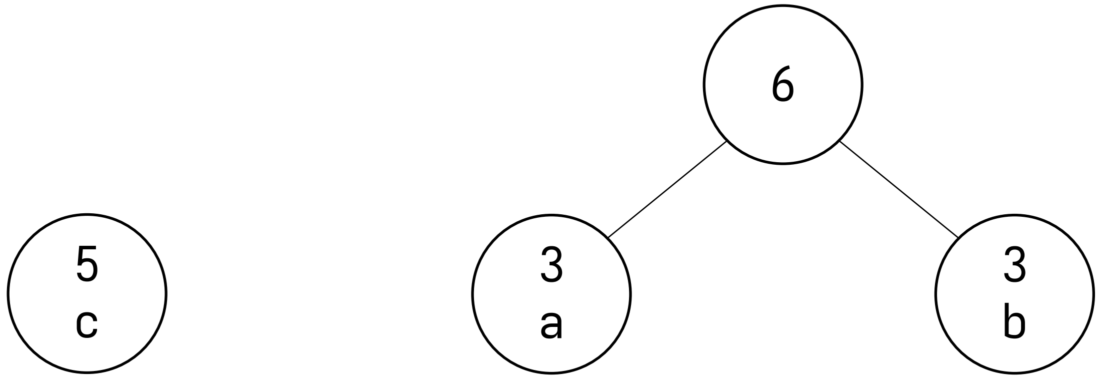
How to do this?

aaabbbccccc



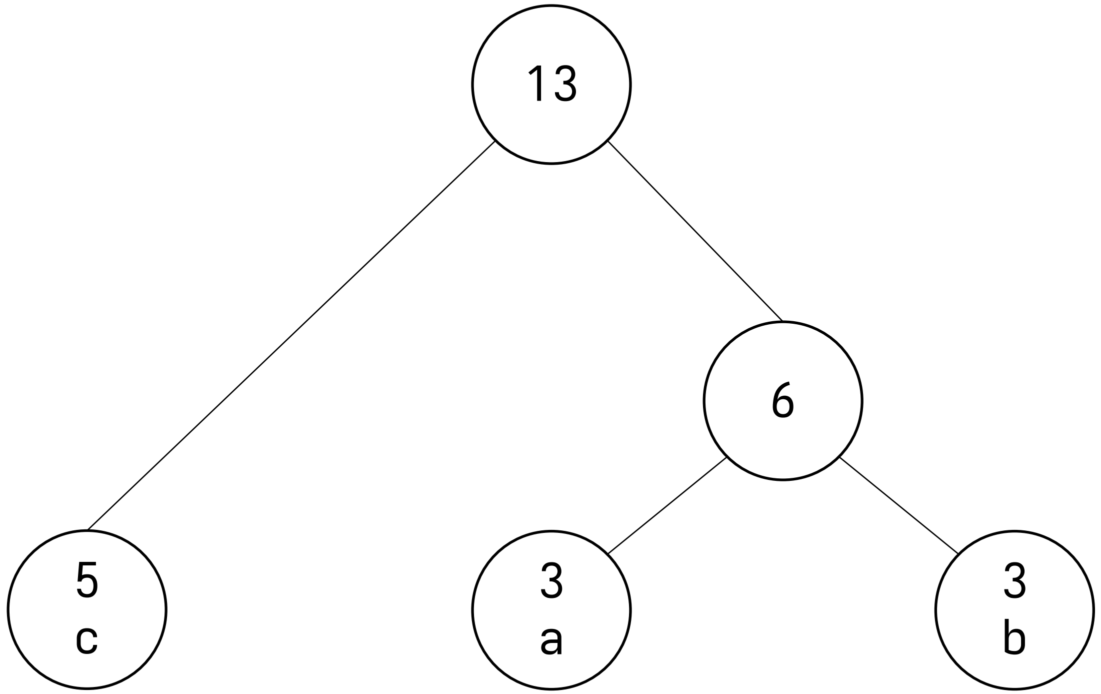
How to do this?

aaabbbccccc



How to do this?

aaabbbccccc



Questions?

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes on Huffman implementation
- 30 minutes for you to explore!
  - Several options:
    - Implement Huffman decoding
    - Learn & implement Lempel-Ziv (LZ77)
    - Combine Huffman & LZ77 to get DEFLATE

The plan

- 5 minutes on RLE theory
- 15 minutes to build RLE
- 15 minutes on Huffman theory
- 15 minutes on Huffman implementation
- **30 minutes for you to explore!**
  - Several options:
    - Implement Huffman decoding
    - Learn & implement Lempel-Ziv (LZ77)
    - Combine Huffman & LZ77 to get DEFLATE

Lempel-Ziv (LZ77)

- Introduced by Lempel and Ziv in 1977
- RLE only removes immediate repetition
- What about this text:

    I like tea, I like cheese, and I like sand.

Lempel-Ziv (LZ77)

I do not like them in a house.
I do not like them with a mouse.
I do not like them here or there.
I do not like them anywhere.

Lempel-Ziv (LZ77)

I do not like them in a house.
I do not like them with a mouse.
I do not like them here or there.
I do not like them anywhere.

Lempel-Ziv (LZ77)

```
I do not like them in a house.
<31,19>with a mouse.
<34,20>here or there.
<35,20>anywhere.
```

The solutions

Fully implemented version of the workbook:

https://puhack.horse/compression-py-solved