# CompSci 161
# Winter 2023 Lecture 18:
# Greedy Algorithms:
# Huffman Compression

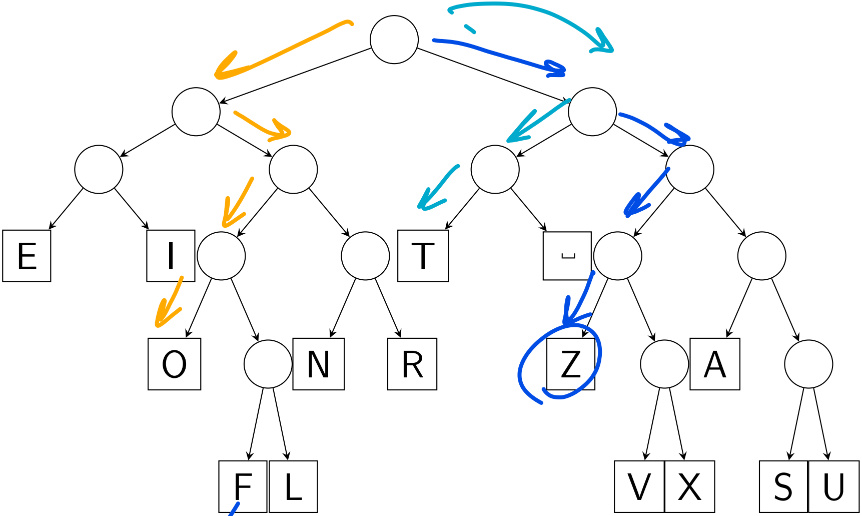
# Candidate Encodings

Suppose we want to encode only letters a . . . z. Identify problems and inefficiencies with the following encodings.

'a' = 96+1 'b' = 96+2 'z' = 96 +26

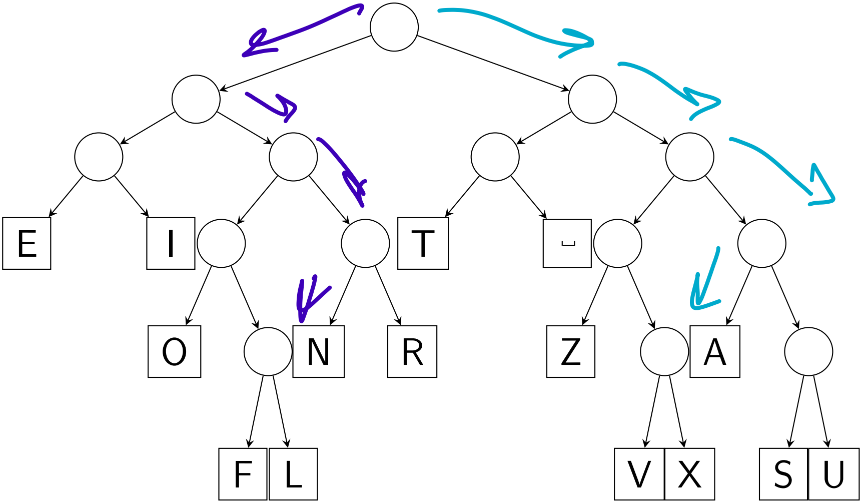
- a = 00000, b = 00001, c = 00010, ..., z = 11001
- a = 0, b = 1, c = 00, d = 01, e = 10, etc
- a = 00000, b = 00001, ..., v = 10101, w = 1100, x = 1101, y = 1110, z = 1111

# One binary tree example




Message: 1100010010010111000100100

# One binary tree example




Message: A N T E A T E R S

1110 0110

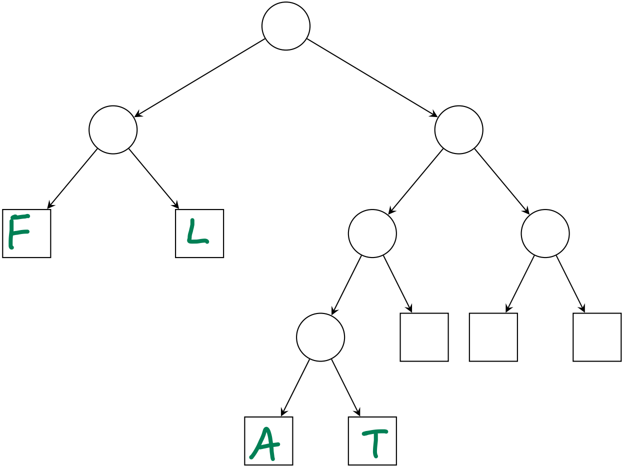
# Why a binary tree? minimize $\sum f_i d_i$

**Lemma 1**: All ~~internal~~ non-leaf nodes in the optimal tree have two children.

FSOC suppose I have an optimal tree T with internal node w with only one child

Create T': T without w. If w was root, T' root is w's child. Else connect w's parent to child... T' has nodes w/ shorter depth (w subtree) and none longer: smaller sum. So T not optimal. ⇐

# Where should the letters go?




| letter | F | I | A | T | L | U | X |
|---|---|---|---|---|---|---|---|
| frequency | 21 | 18 | 6 | 5 | 23 | 12 | 15 |

# Why least frequent at max depth?

i.e. : diff depths, lower frequency ⟷ deeper

**Lemma 2**: The two characters with minimum frequency should be at maximum depth

FSOC suppose $c$ and $e$ $d_c < d_e$ but $f_c < f_e$. Swap $c$ and $e$.

Change in $\sum f_i d_i$ ?

$$+ f_c (d_e - d_c) - f_e(-d_c + d_e)$$

$$(f_c - f_e)(d_e - d_c)$$

$<0$ $>0$

So post-swap has better cost.

7 Let's build a tree for "`engineering useless rings`"

Step one: count the characters.

| char | count |
|---|---|
| e | |
| n | |
| g | |
| i | |
| r | |
| ␣ | |
| u | |
| s | |
| l | |

# Let's build a tree for "engineering useless rings"

Step two : Create leaf nodes and then build the tree.

| char | count |
|---|---|
| e | 5 |
| n | 4 |
| s | 4 |
| g | 3 |
| i | 3 |
| r | 2 |
| ␣ | 2 |
| u | 1 |
| l | 1 |