Computer Science
Spring 2021 Lecture 22:
Greedy Algorithms:
Huffman Compression

2 Candidate Encodings

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

- a = 00000, b = 00001, c = 00010, ..., z = 11001

  11100

- a = 0, b = 1, c = 00, d = 01, e = 10, etc

  601 ad? aab?

- a = 00000, b = 00001, ..., v = 10101, w = 1100, x = 1101, y = 1110, z = 1111

3 Using a tree for an encoding?

How could we use a binary tree to represent an encoding?

- Ensure no symbol has another as prefix?

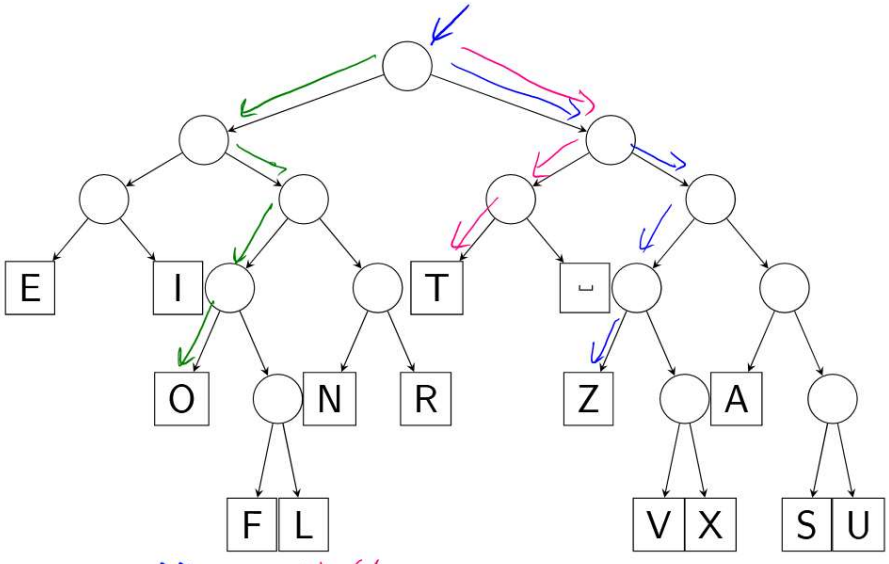
  encoding: path root → leaf

- What part(s) of the tree should be symbols?

  leaf nodes

- What do left and right children mean?

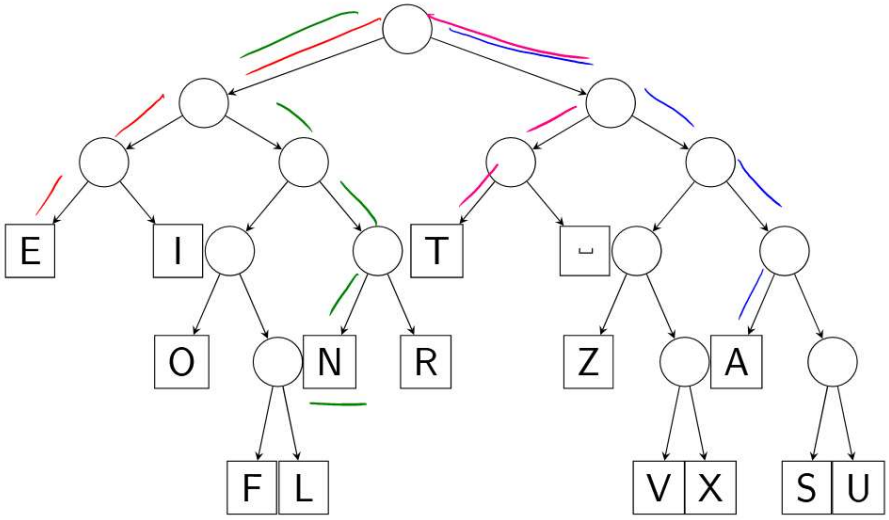
4 One binary tree example



Message: 1100010011001011100010100

Z O T

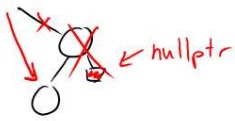
4 # One binary tree example



Message: A N T E A T E R S

1110 0110 100 000 1110

5 # Why a binary tree?



Lemma 1: All internal nodes in the optimal tree have two children.

Suppose FSOC some optimal tree T has a node w with only one child

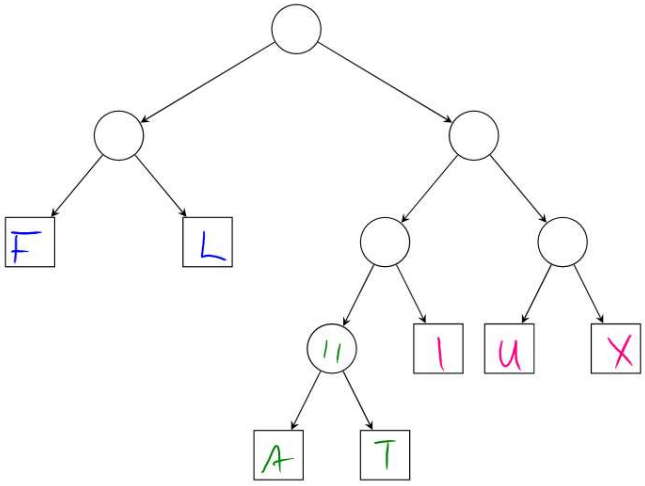
Create new tree T': copy T, with w removed. Its parent points to its child instead. ...

---

our goal: minimize $\sum f_i b_i$: total bits used to encode message

6

Where should the letters go?



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

7

Why least frequent at max depth?

Call the two least frequent c, d

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

Suppose FSOC c not at max depth and e at max depth, $e \neq c$, $e \neq d$

$f_e > f_c$ $d_e > d_c$ ← depth; length of encoding

If e and c are swapped, what is change in cost?

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

$= (f_c - f_e)(d_e - d_c)$: change is

<0 >0 negative, so cost decreases ▨

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

Step one: count the characters.

char	count
e	
n	
g	
i	
r	
␣	
u	
s	
l	

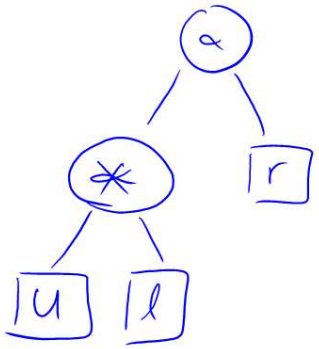
We will do this on Wednesday at start of class.

9 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~~

α 4



10

Why is this optimal?

Goal is to minimize $\sum_i f_i b_i$