Learning Goals

- Describe "Binary code," "Prefix free", "Average letter length"
- Explain connection between binary codes + trees
- Describe Huffman's alg.
- Analyze runtime of Huffman's alg
- Describe impact of data structures alg runtime
- Prove correctness of Huffman's alg

Binary Codes

def:

ex:

Suppose you have a message where the letter "a" occurs 50% of the time, "b" 30%, and "c" 20%. Which is the best binary encoding of $\Sigma = \{a, b, c\}$?

A): $f(a) = 00$, $f(b) = 01$, $f(c) = 10$

B) $f(a) = 0$, $f(b) = 1$, $f(c) = 01$

C) $f(a) = 0$, $f(b) = 10$, $f(c) = 11$

A): $f(a) = 00$, $f(b) = 01$, $f(c) = 10$

B) $f(a) = 0$, $f(b) = 1$, $f(c) = 01$

C) $f(a) = 0$, $f(b) = 10$, $f(c) = 11$

Average letter length:

ex:

Binary Trees & Binary Codes

$f(a) = 0$

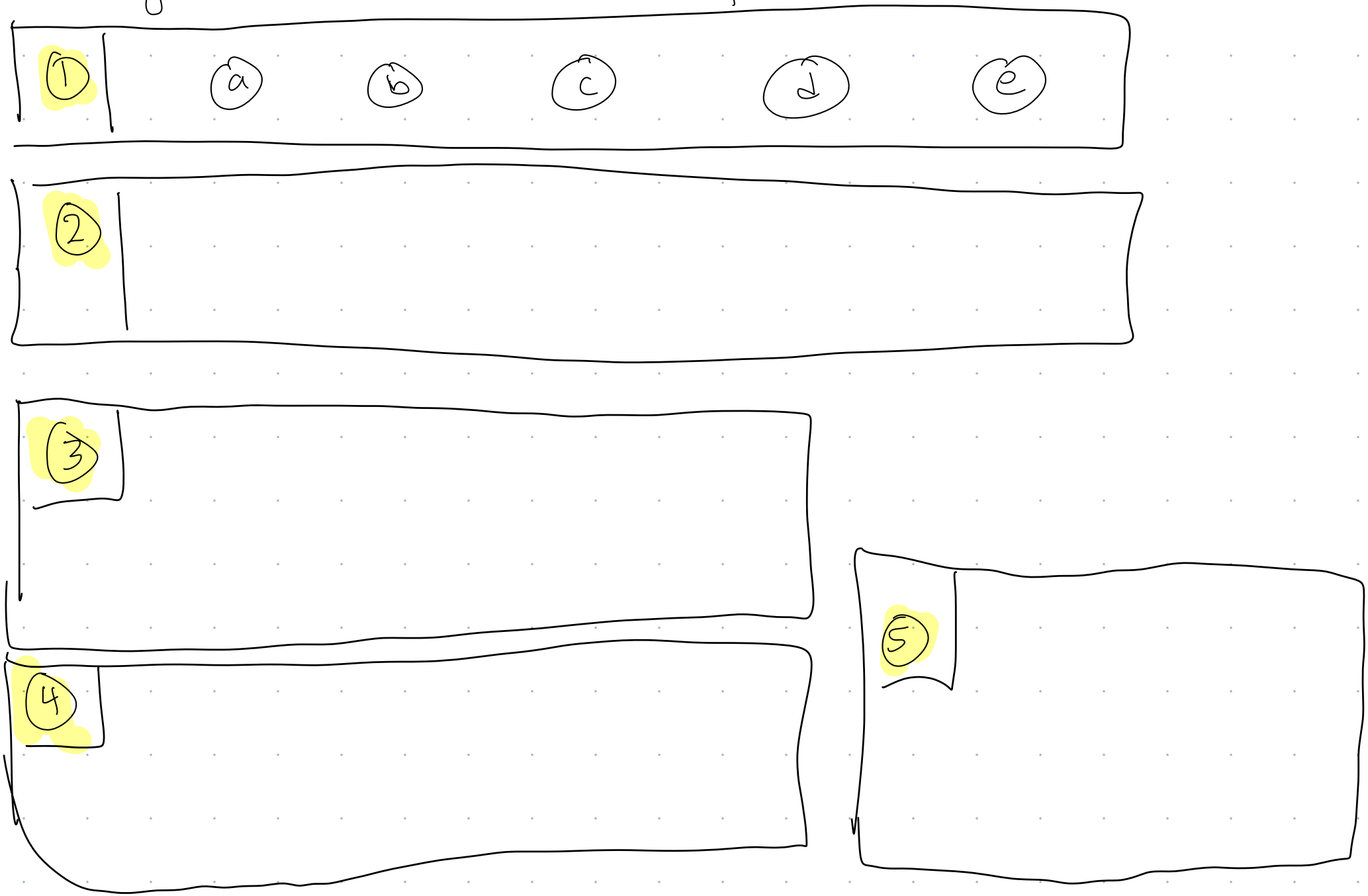
$f(b) = 01$

$f(c) = 11$

$\longleftrightarrow$

def:

Merge Trees to Create Prefix Free Codes



Optimal Binary Encoding Problem

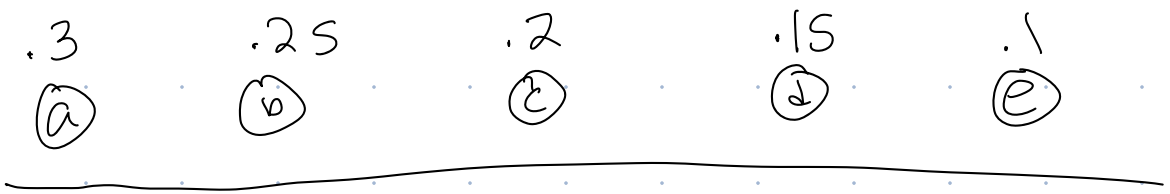
Input:

Output:

Huffman's Algorithm

i	p(i)
a	.3
b	.25
c	.2
d	.15
e	.1

- Use Huffman's algorithm to create a binary code
- What is the average letter length of your code
- What is the runtime of Huffman's in terms of $|\Sigma| = n$? Ideas to improve?
- Why greedy?


.3
.25
.2
.15
.1
a
b
c
d
e

Huffman's Algorithm

For each $i \in \Sigma$:

- Create a tree with one node, label i
- Give tree weight $p(i)$

While there is more than one tree:

- Merge two trees with smallest weights
- Set weight of merged tree to be sum of weights

Why greedy:

Huffman's Algorithm

For each $i \in \Sigma$:

- Create a tree with one node, label i
- Give tree weight $p(i)$

While there is more than one tree:

- Merge two trees with smallest weights
- Set weight of merged tree to be sum of weights

Improve runtime?

- At each iteration of while loop, find minimum value tree.
- Helpful data structure??

Go Program! (Lots of details to figure out!)

Ethical Matrix (see programming assignment)

Thm: Huffman's Algorithm produces a prefix free code that minimizes average letter length.

Pf:

ex:

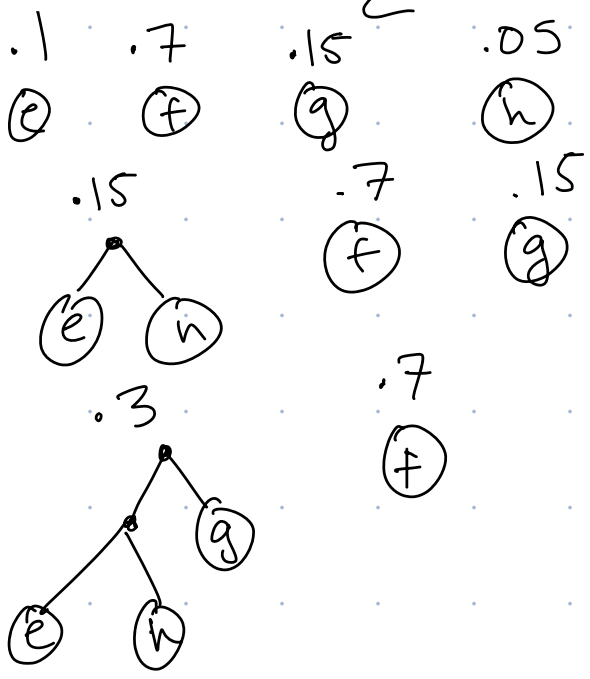
$\Sigma = \{e, f, g, h\}$ $p(e) = .1$ $p(f) = .7$ $p(g) = .15$ $p(h) = .05$

then

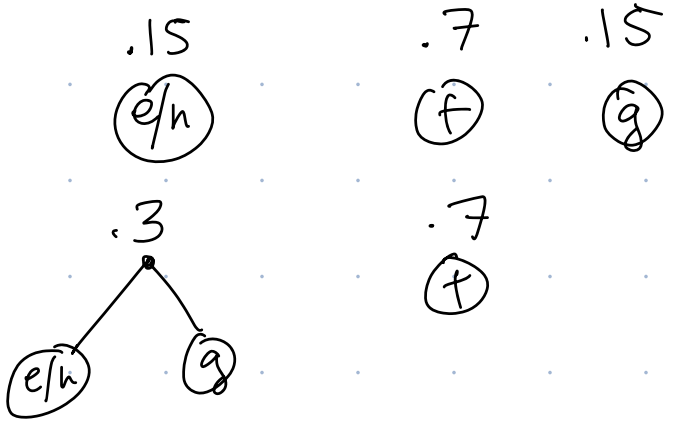
$\Sigma^- =$

Huffman

Σ



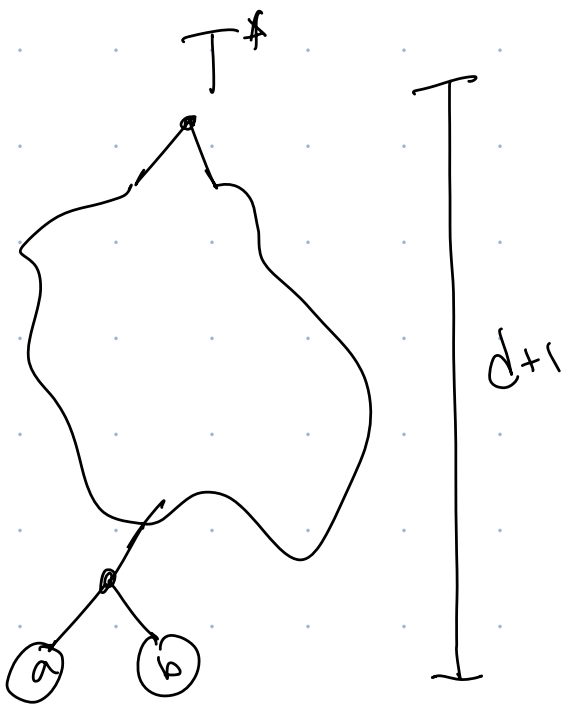
Σ^-



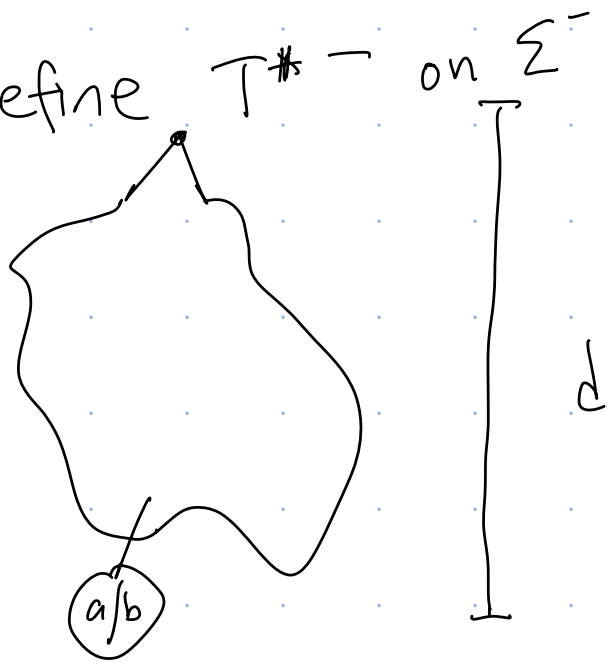
In general:

Huffman

Lemma: There is an optimal tree for Σ where a, b are siblings. (will prove later)



Define T^*_- on Σ^-



(fill in with d, $p(a)$, $p(b)$)

Then

$$L(T^*) = \sum_{i \neq a,b \in \Sigma} p(i)\,d(i) \quad + \quad \square$$

$$L(T^*_-) = \sum_{i \neq a/b} p(i)\,d(i) \quad + \quad \square$$

So $L(T^*) - L(T^{*-}) =$ [ ]

Similarly

$L(T) - L(T^-) =$ [ ]

Thus

$$L(T^*) - L(T^{*-}) = L(T) - L(T^-)$$

Rearranging: $L(T) - L(T^*) =$ [ ]

This is a contradiction because

[ ]

Lemma: There is an optimal tree for Σ with a, b (characters with smallest p-values) siblings.