

PRIORITY QUEUE : APPLICATIONS.

1. SELECTING K^{th} SMALLEST ELEMENT

2. CODING.

SELECTION PROBLEM.

Given n elements - compute the
 k^{th} smallest element

INPUT : n integers and an integer k
 $1 \leq k \leq n$

OUTPUT : k^{th} smallest element in the
input

Special cases : $k = 1$
 $k = 2$

SELECTION PROBLEM.

Possible approaches

1. Sort the input, return $A[k]$

2. Sort k elements, process remaining $n-k$ elements appropriately

7, 9, 3, 10, 12, 1, 8, 2



SELECTION PROBLEM.

Using a priority queue

- create a heap with n elements
- perform k delete min operations.

$$(n + k \cdot \log n)$$

SELECTION PROBLEM.

APPROACHES

1. $O(n \log n)$

2. $O(n \cdot k)$

3. $O(n + k \log n)$

PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

CHAR	FREQ
------	------

e	12
---	----

t	9
---	---

a	8
---	---

o	7
---	---

i	7
---	---

PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

~ Prefix codes

CHAR	FREQ	C1	C2	C3	
					0001
e	12	0	0	10	0.001 : eat
t	9	1	10	01	0001 : aet
a	8	00	110	00	
o	7	01	1110	110	
i	7	10	1111	111	

PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

CHAR	FREQ	C1	C2	C3
e	12	0	0	10
t	9	1	10	01
a	8	00	110	00
o	7	01	1110	110
i	7	10	1111	111

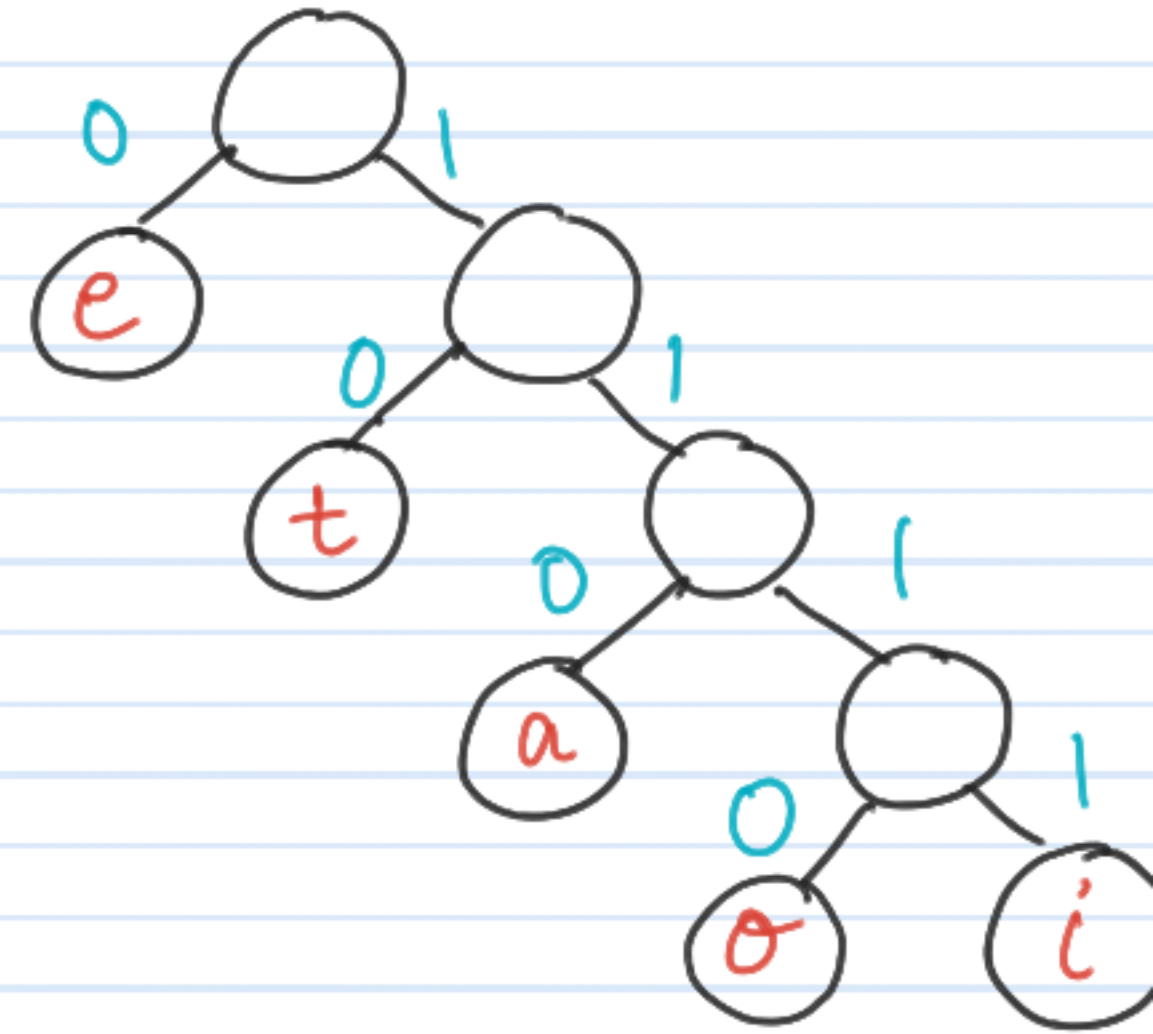
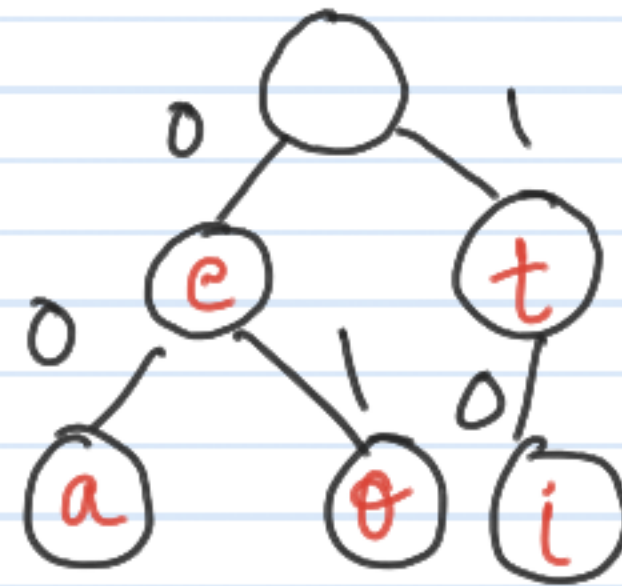
Goals:

- Higher freq \Rightarrow shorter code
- unambiguous decoding
- Total code length (optimize)

PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

CHAR	FREQ	C1	C2	C3
e	12	0	0	10
t	9	1	10	01
a	8	00	110	00
o	7	01	1110	110
i	7	10	1111	111



PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

CHARACTER FREQUENCY

e	12	
t	9	} 17
a	8	
o	7	} 14
i	7	

Diagram description: A tree structure is shown with a root node '26' on the left. A vertical line descends from '26' and splits into two horizontal branches. The upper branch leads to a node '12' (character 'e'). The lower branch leads to a node '17'. From '17', a vertical line descends and splits into two horizontal branches leading to nodes '9' (character 't') and '8' (character 'a'). From '8', a vertical line descends and splits into two horizontal branches leading to nodes '7' (character 'o') and '7' (character 'i').

algo to generate code

- pick two smallest frequencies
- combine them to form a new freq (dummy char)
- Remove the 2 freq add the new one
- Repeat

PRIORITY QUEUE : APPLICATION 2

HUFFMAN CODING.

CHARACTER FREQUENCY

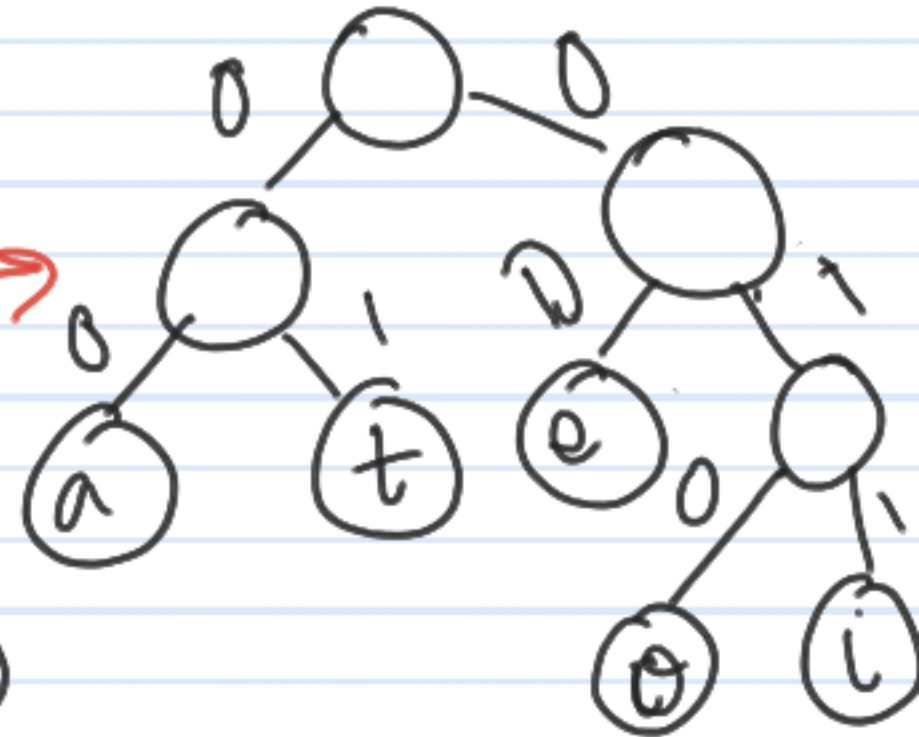
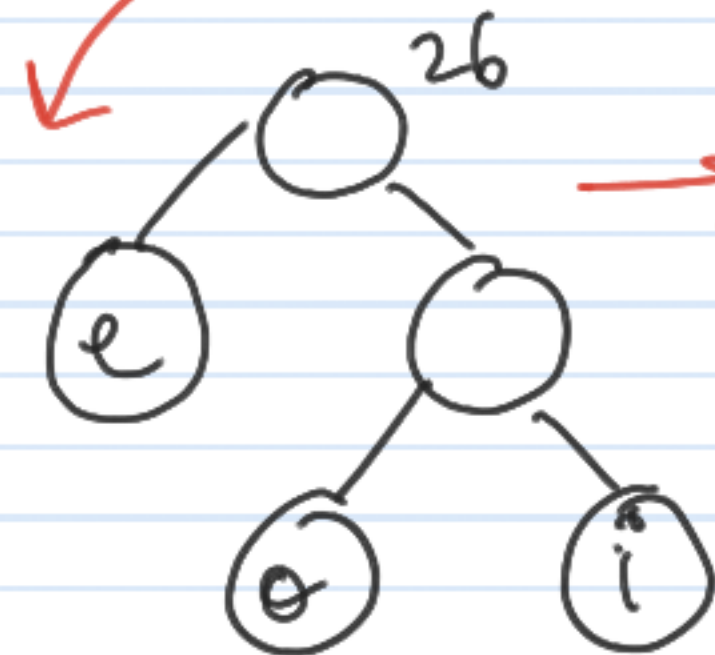
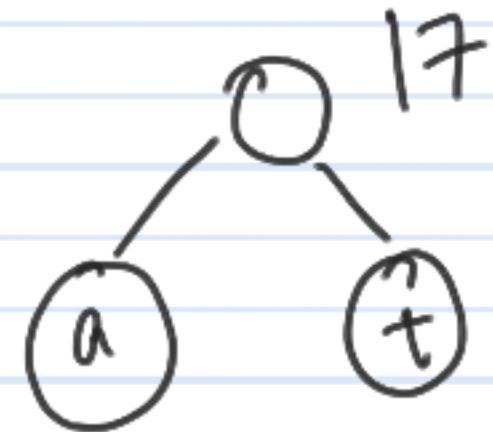
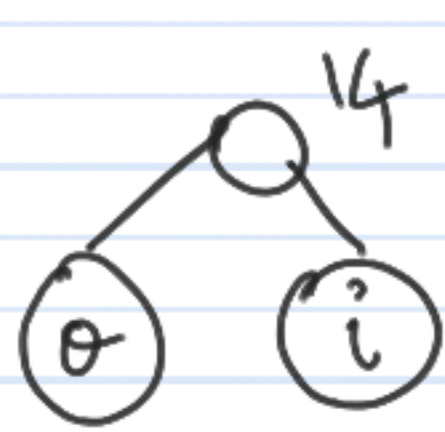
e 12

t 9

a 8

o 7

i 7



C_3