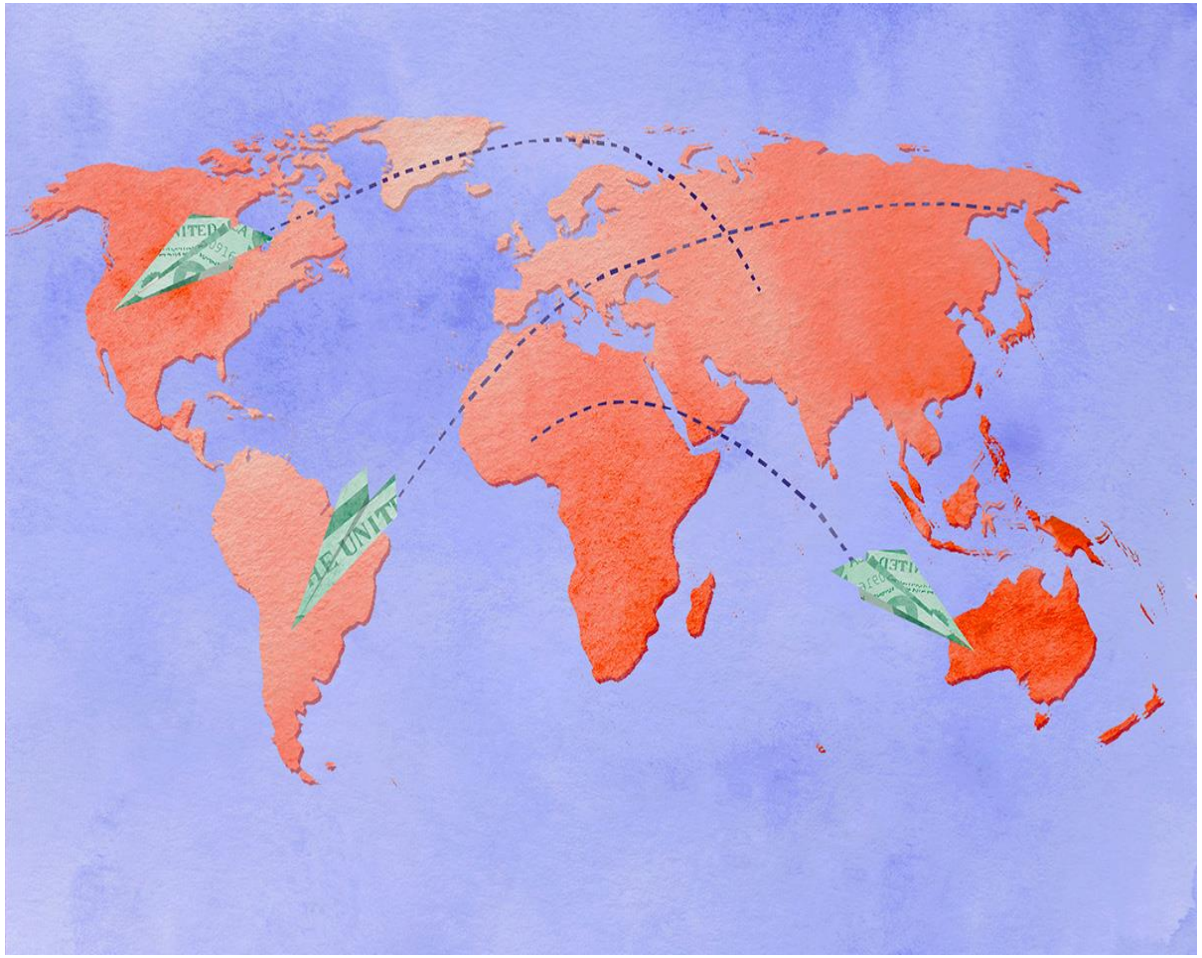
https://forms.gle/KPCqdw4dLaTDxkGy5

556755

Greedy Lecture 2



Two researchers from the Massachusetts Institute of Technology and a third from Harvard University won the 2019 Nobel Prize in economics on Monday for groundbreaking research into what works and what doesn't in the fight to reduce global poverty.

The award went to MIT's Esther Duflo and Abhijit Banerjee, and Harvard's Michael Kremer. The 46-year-old Duflo is the youngest person ever to win the prize and only the second woman, after Elinor Ostrom in 2009.

The three winners, who have worked together, revolutionized developmental economics by pioneering field experiments that generate practical insights into how poor people respond to education, health care and other programs meant to lift them out of poverty.

e: 74	***f: 11***	***I: 2***
o: 55	***g: 11***	***H: 2***
t: 50	***u: 10***	***E: 2***
r: 48	***y: 9***	***D: 2***
n: 43	***v: 9***	***j: 2***
a: 38	***T: 6***	***-: 2***
i: 33	***b: 5***	***1: 1***
h: 31	***,: 5***	***4: 1***
s: 27	***M: 4***	***6: 1***
d: 22	***.: 4***	***U: 1***
l: 18	***↵: 4***	***N: 1***
c: 15	***0: 3***	***P: 1***
w: 13	***z: 3***	***A: 1***
p: 13	***k: 3***	***B: 1***
m: 12	***’: 3***	***K: 1***
	2: 2	***O: 1***
	9: 2	***x: 1***

Two researchers from the Massachusetts Institute of Technology and a third from Harvard University won the 2019 Nobel Prize in economics on Monday for groundbreaking research into what works and what doesn’t in the fight to reduce global poverty.

The award went to MIT’s Esther Duflo and Abhijit Banerjee, and Harvard’s Michael Kremer. The 46-year-old Duflo is the youngest person ever to win the prize and only the second woman, after Elinor Ostrom in 2009.
The three winners, who have worked together, revolutionized developmental economics by pioneering field experiments that generate practical insights into how poor people respond to education, health care and other programs meant to lift them out of poverty.

$c \in A$	f_c	T
e:	74	000
o:	55	001
t:	50	010
r:	48	011
n:	43	100
a:	38	101
	308	

$c \in A$	f_c	T	l_c	
e:	74			
o:	55			
t:	50			
r:	48			
n:	43			
a:	38			
	308		3	924

$c \in A$	f_c	T	l_c	
e:	**74**	000	3	
o:	**55**	001	3	
t:	**50**	010	3	
r:	**48**	011	3	
n:	**43**	100	3	
a:	**38**	101	3	
	308		**3**	**924**

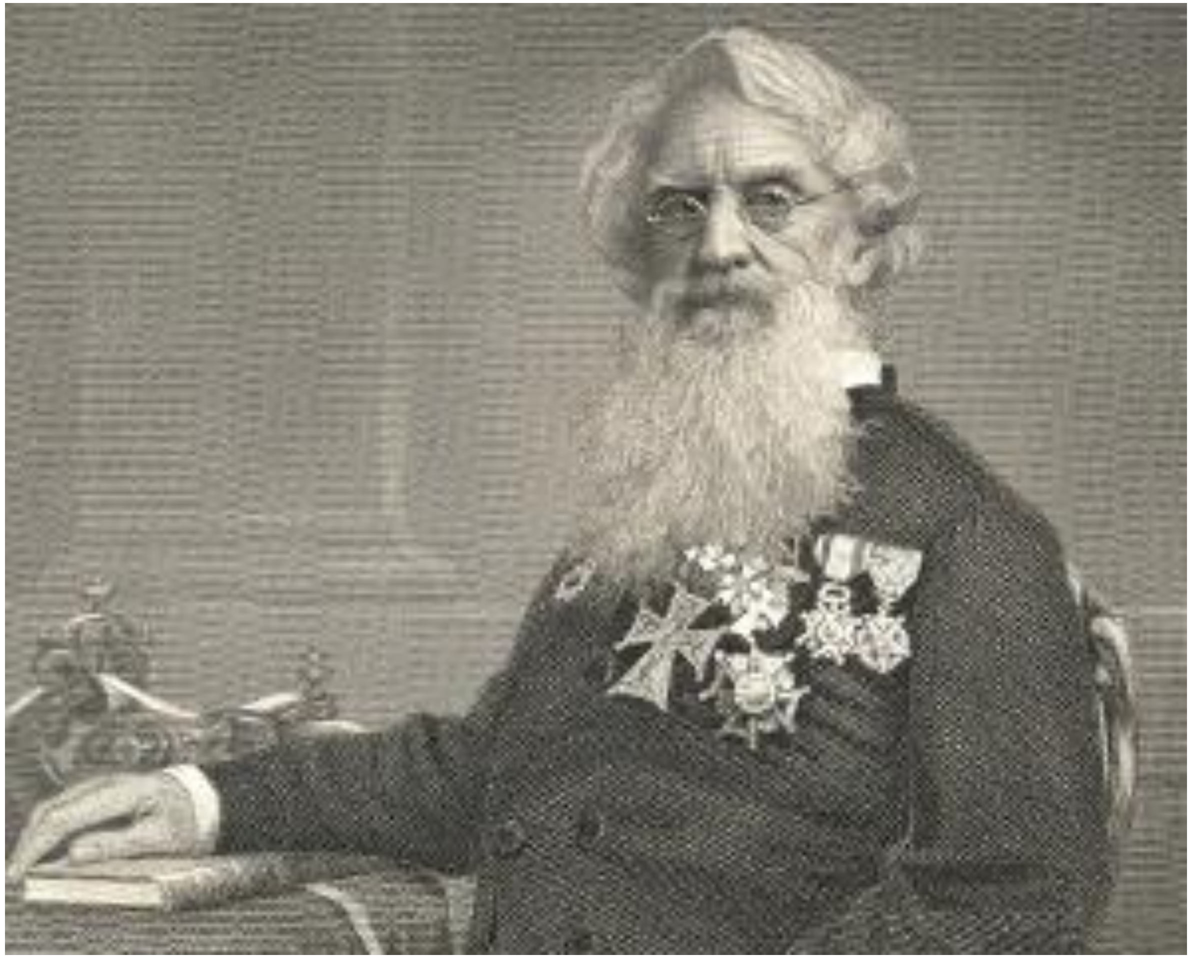
Cost of an encoding

$c \in A$	f_c	T	l_c
e:	74	000	3
o:	55	001	3
t:	50	010	3
r:	48	011	3
n:	43	100	3
a:	38	101	3
	308		3

$$B(T,\{f_c\}) = \sum_{c \in A} f_c l_c$$

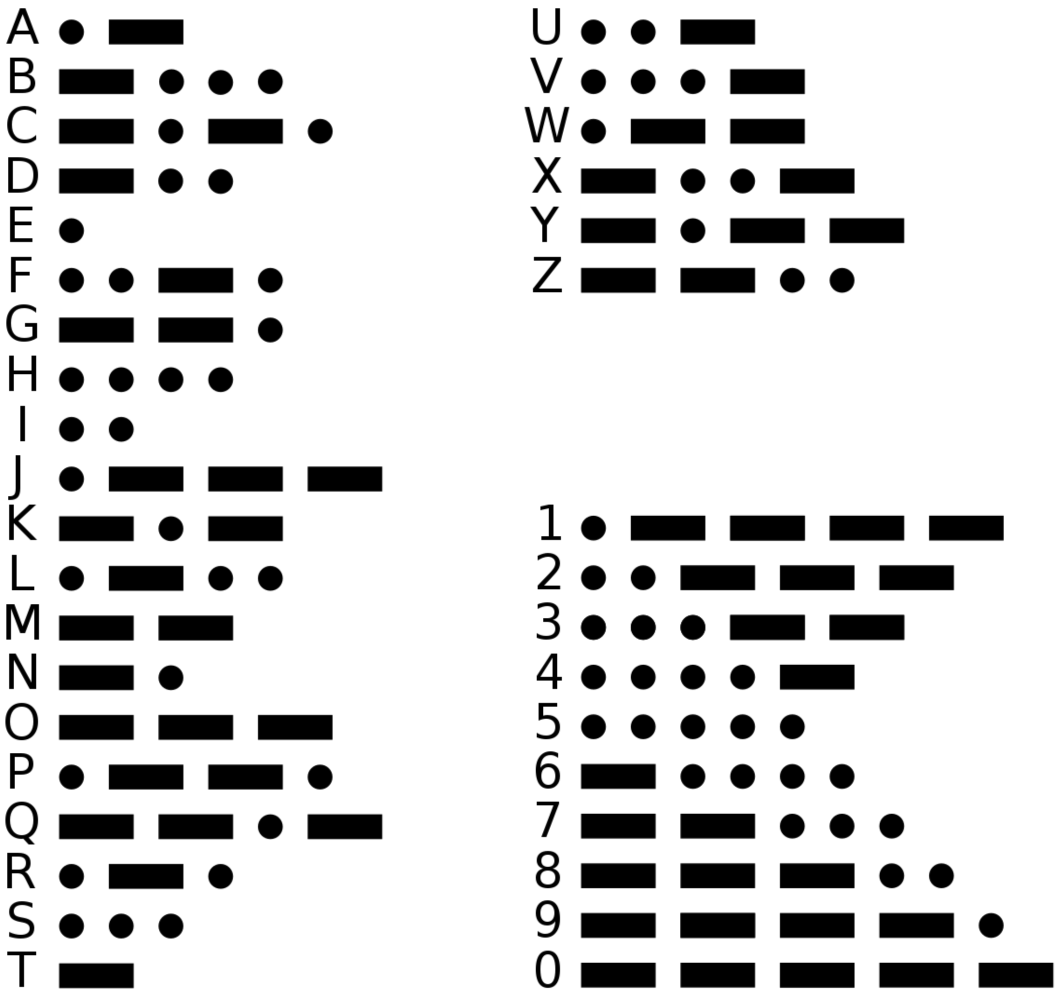
924

Morse Code



International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.



A A

E T E T

E is the prefix of encoding for A

Code can be decoded to several messages

Prefix-free code

Code for alphabets A such that for any two symbol $x,y \in A$, if $x \neq y$ then code (x) is not a prefix of code(y)

Prefix-free code

$c \in A$	f_c	T
e:	**74**	0
o:	**55**	10
t:	**50**	110
r:	**48**	1110
n:	**43**	11110
a:	**38**	111110
	308	

Prefix-free code

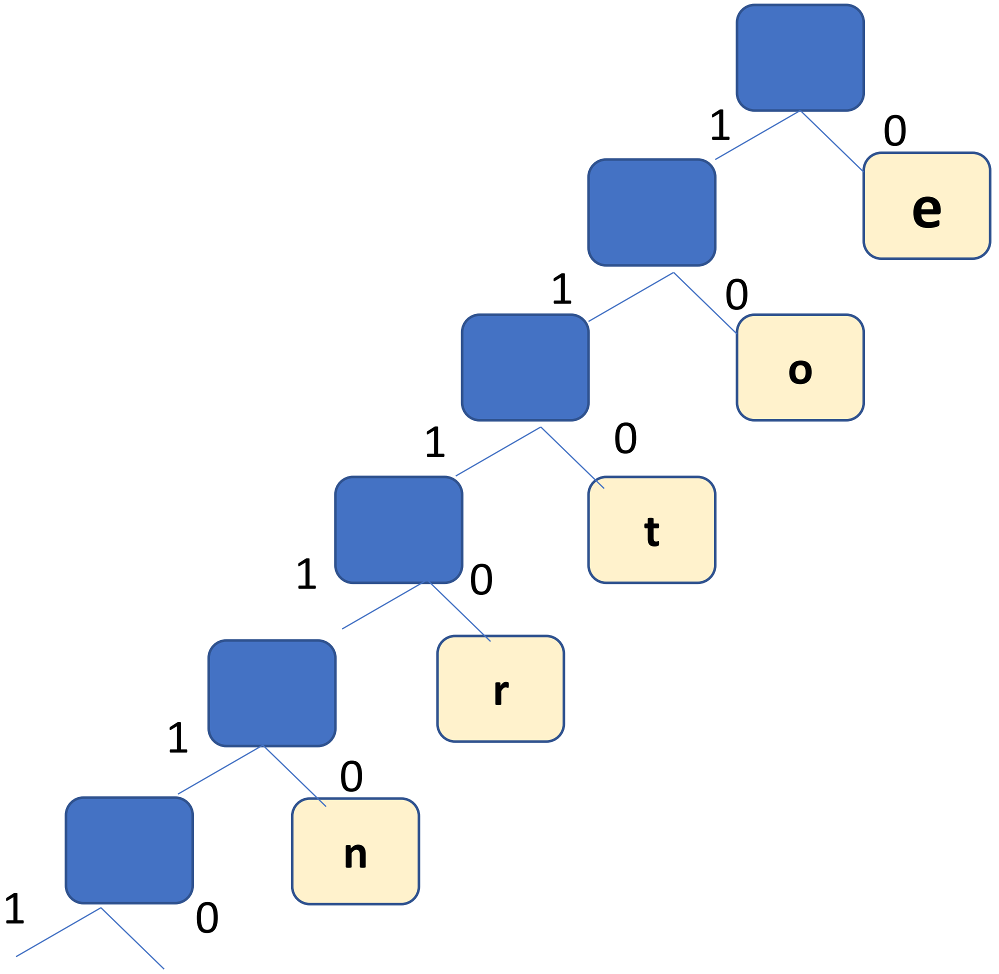
$c \in A$	f_c	T
e:	**74**	0
o:	**55**	10
t:	**50**	110
r:	**48**	1110
n:	**43**	11110
a:	**38**	111110
	308	

11110101111100

Prefix-free code

11110101111100

c ∈ A	f_c	T
e:	74	0
o:	55	10
t:	50	110
r:	48	1110
n:	43	11110
a:	38	111110
	308	



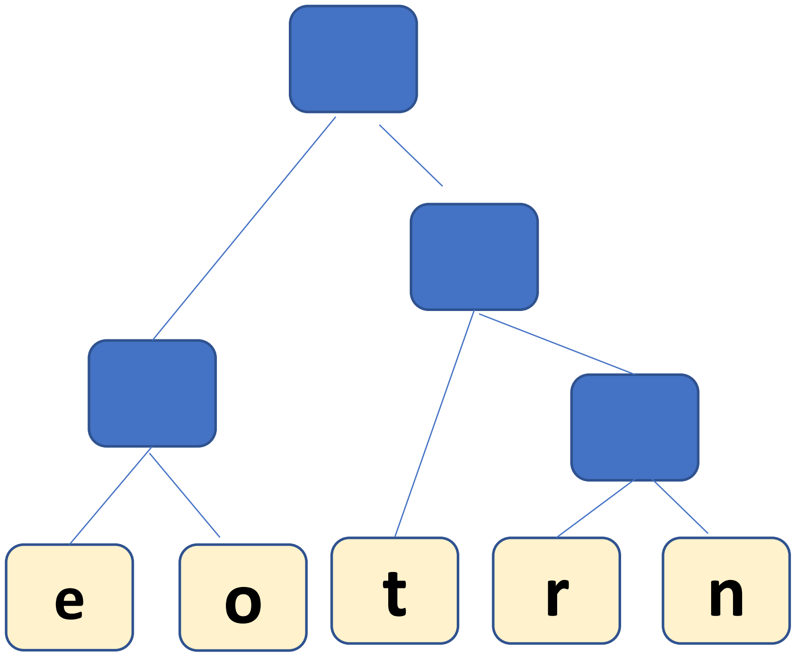
Prefix-free code

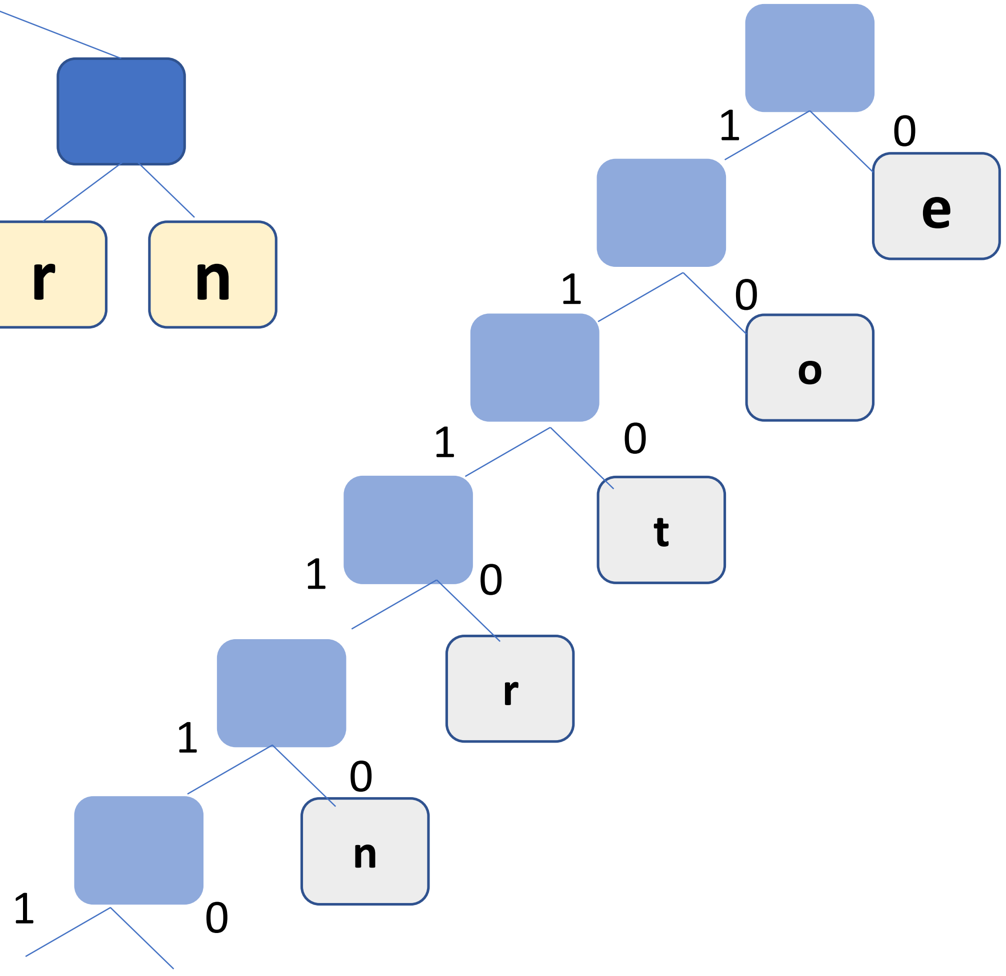
$\Updownarrow$

Binary tree

Prefix-free code

$c \in A$	f_c	T
e:	74	0
o:	55	10
t:	50	110
r:	48	1110
n:	43	11110

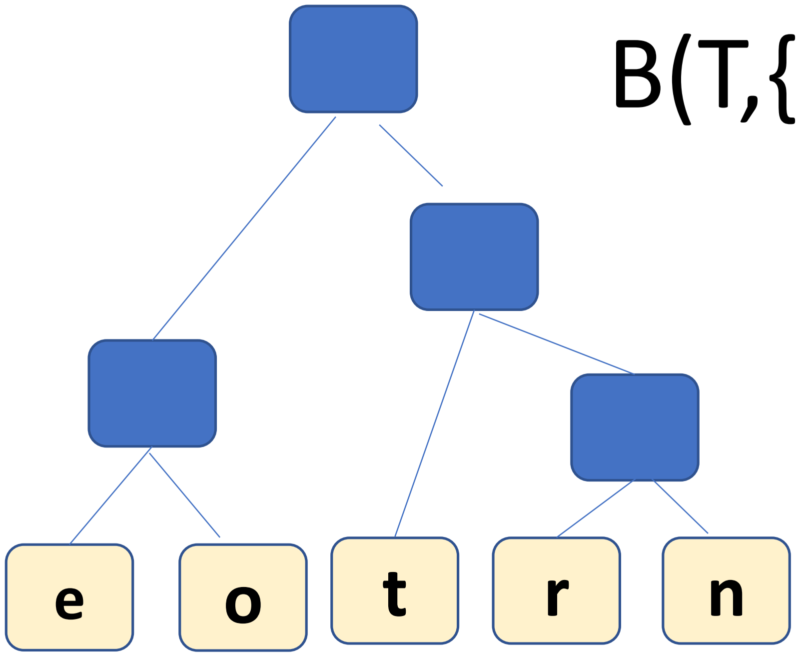


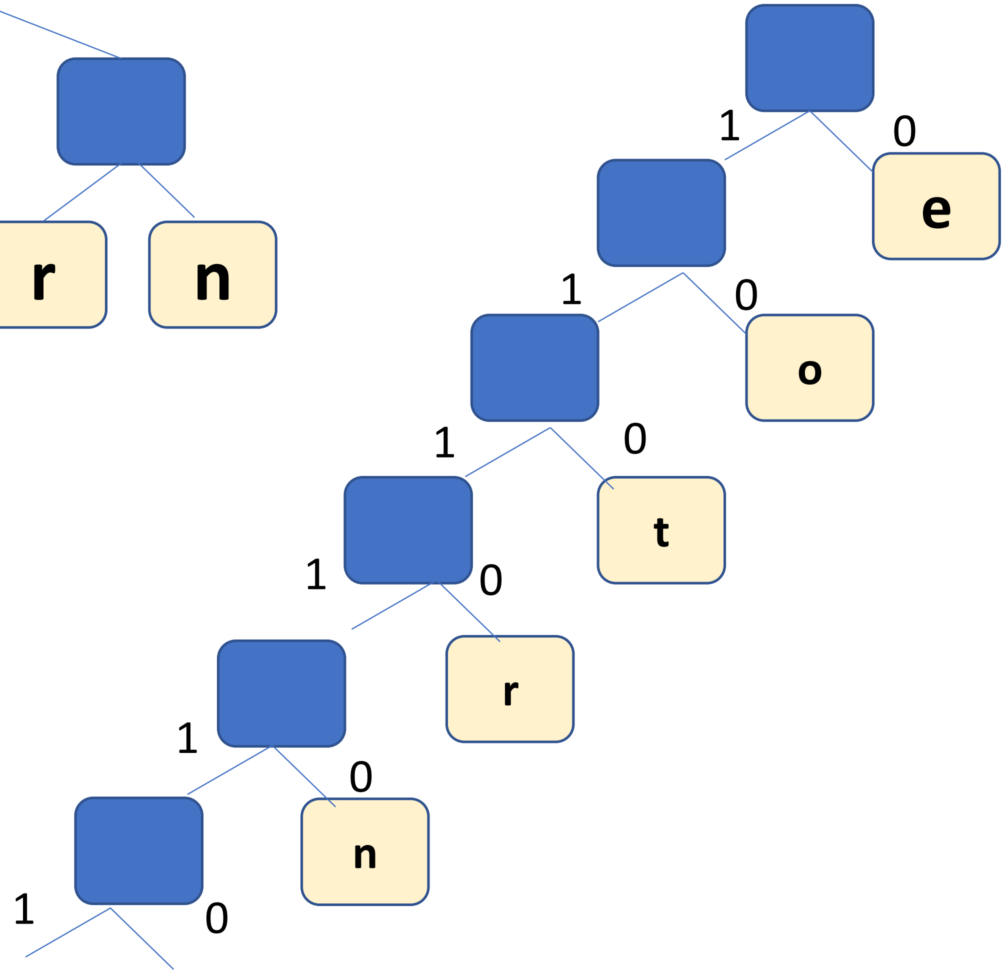


Prefix-free code

$$B(T,\{f_C\}) = \sum_{\mathbf{c} \in \mathbf{A}} f_c l_c$$

c ϵ A	f_c	T
e:	**74**	00
o:	**55**	01
t:	**50**	10
r:	**48**	110
n:	**43**	111





Goal

given the character frequencies $\{f_c\}_{c \in \mathsf{A}}$

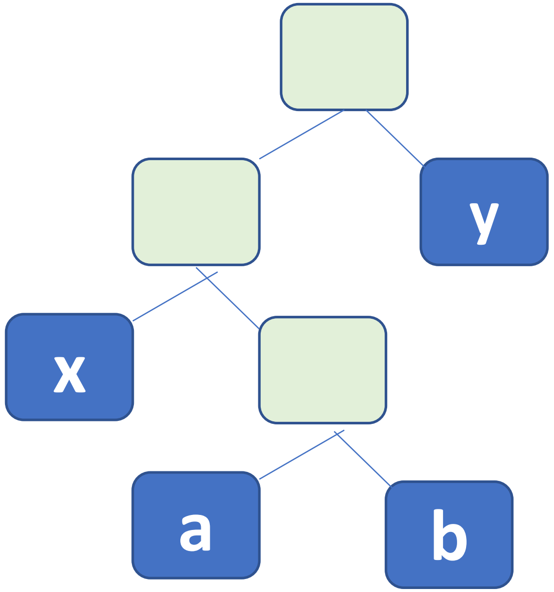
produce a prefix code T with smallest cost

$$\min_T B(T, \{f_c\})$$

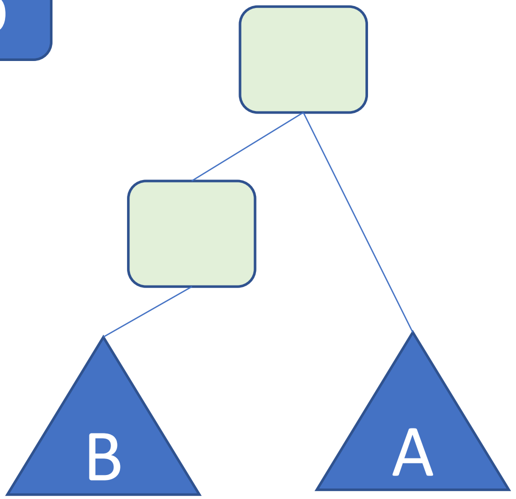
Property

Lemma: Optimal tree is full

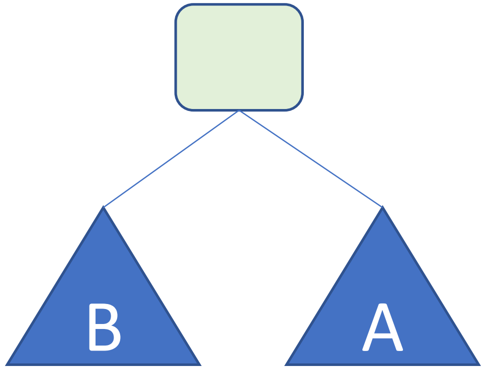
Every node in the optimal tree has either 0 or 2 children.



T



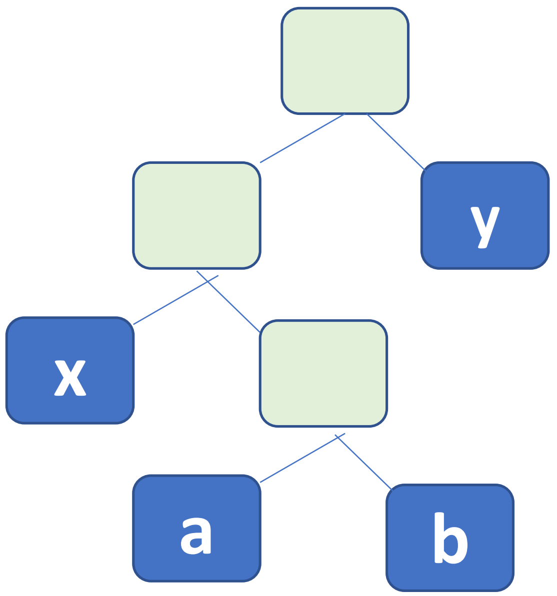
T’



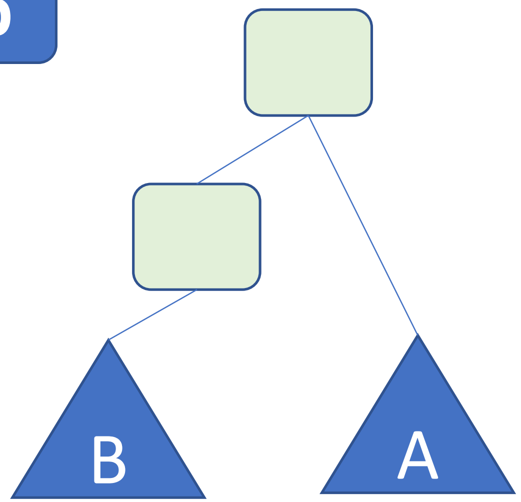
Property

Lemma: Optimal tree is full

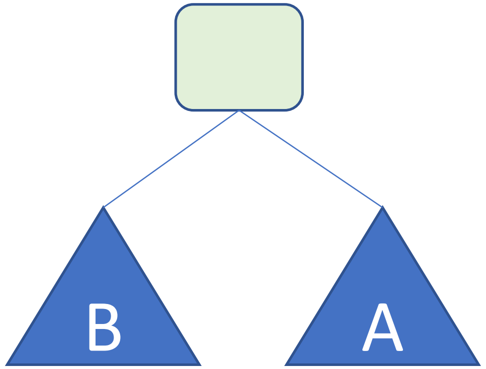
Every node in the optimal tree has either 0 or 2 children.



T



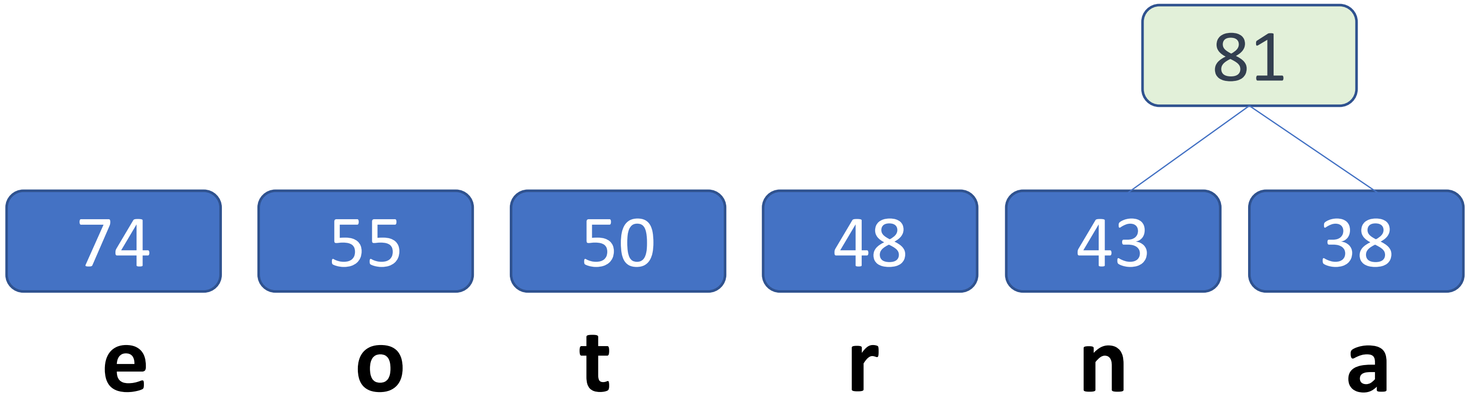
T’



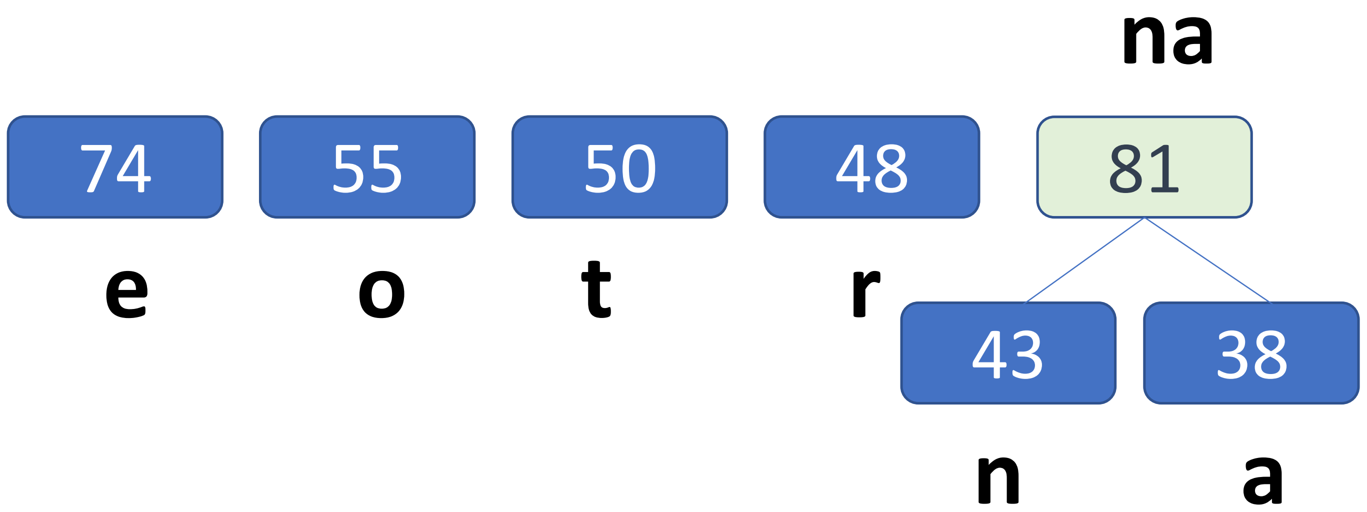
Huffman coding

$c \in A$	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38

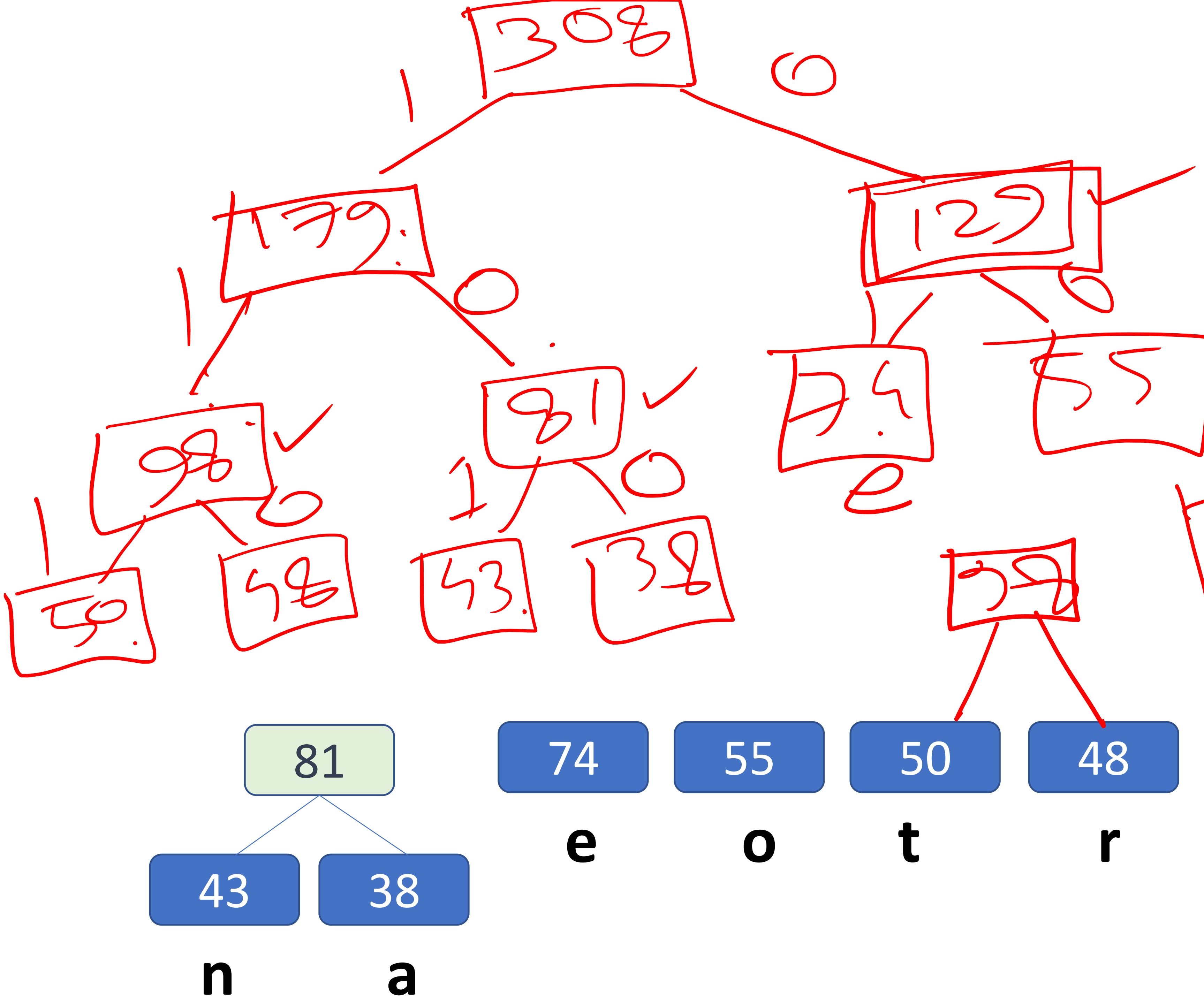
$c \in A$	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



$c \in A$	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



c ∈ A	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



e = 01

o = 00

~~N~~-1 111

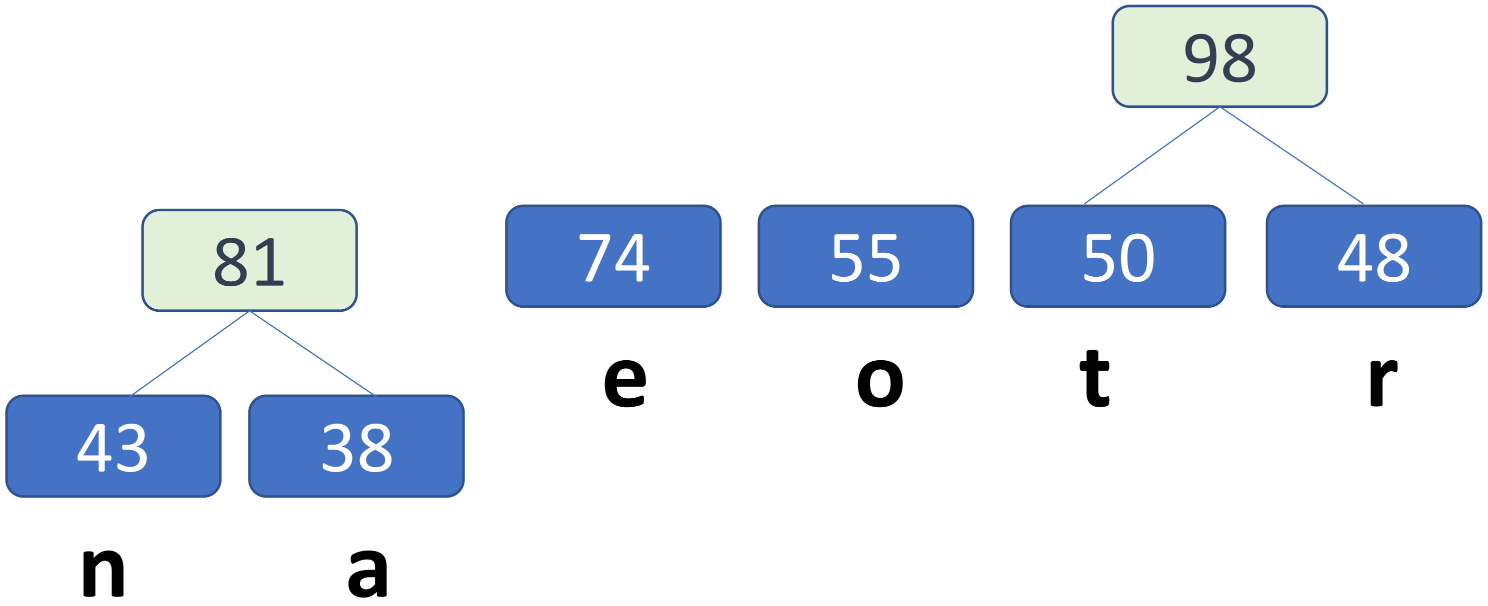
character in A

r = 110

n = 101

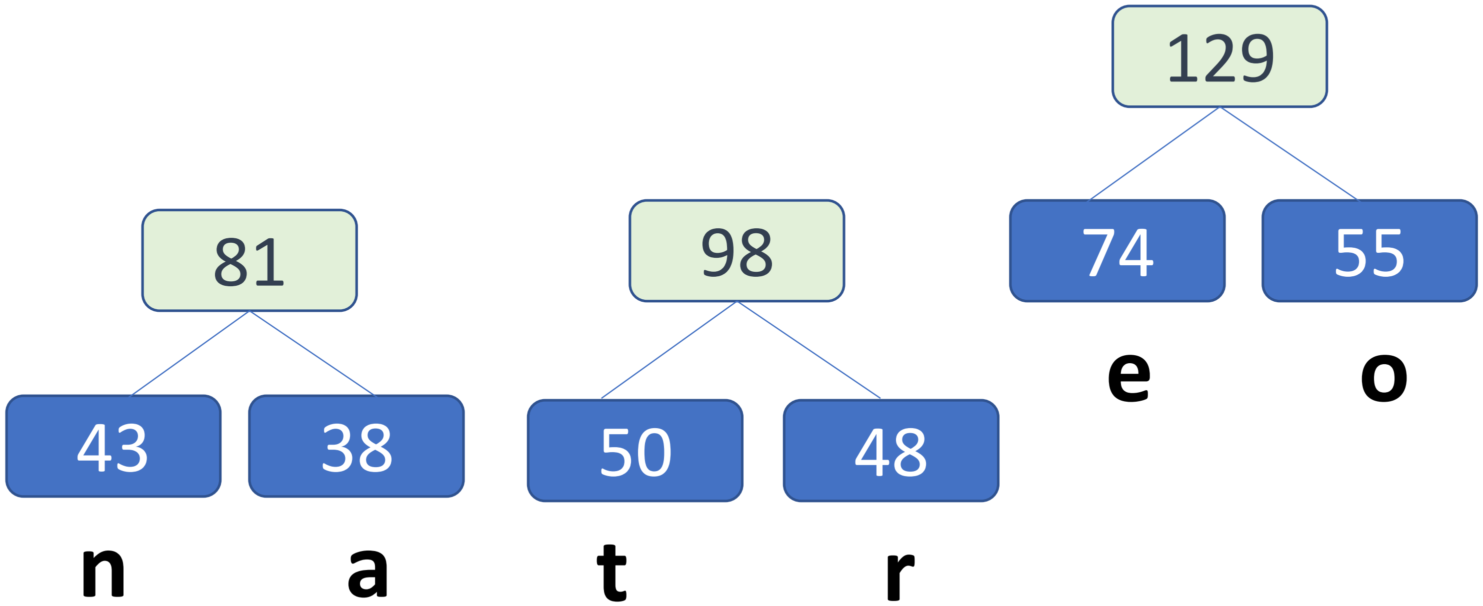
a = 100

$c \in A$	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



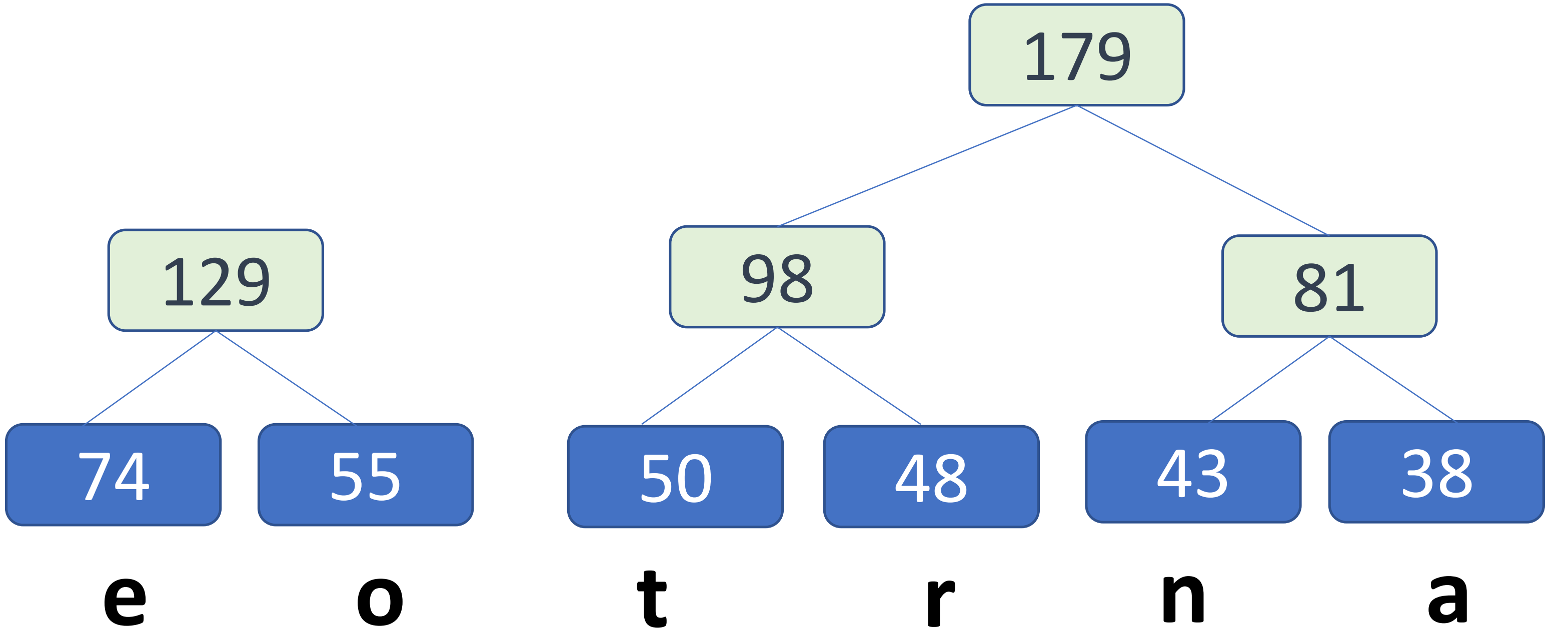
N-1 character in A

c ϵ A	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



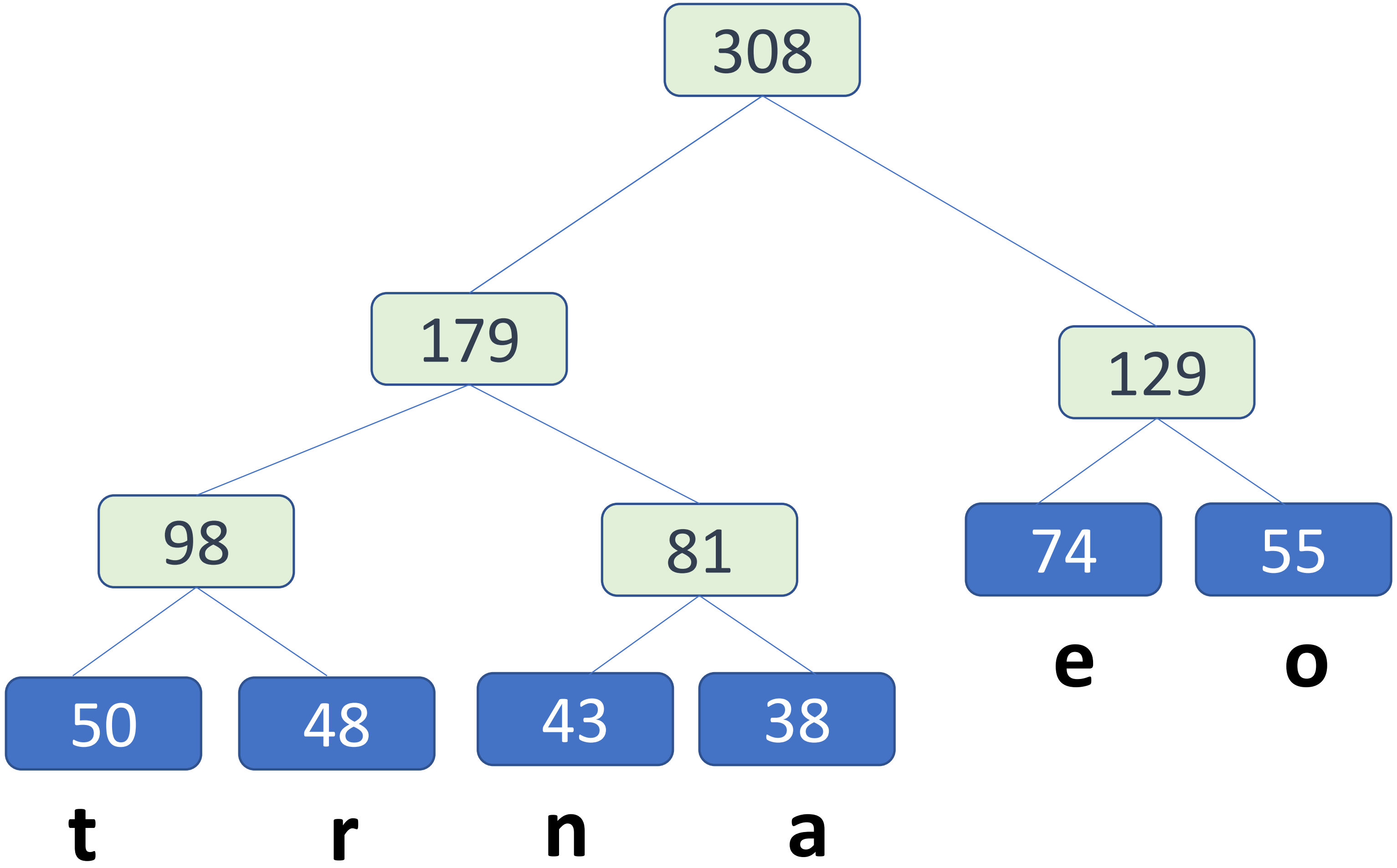
N-2 character in A

c ϵ A	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



N-3 character in A

$c \in A$	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



N-4 characte r in A

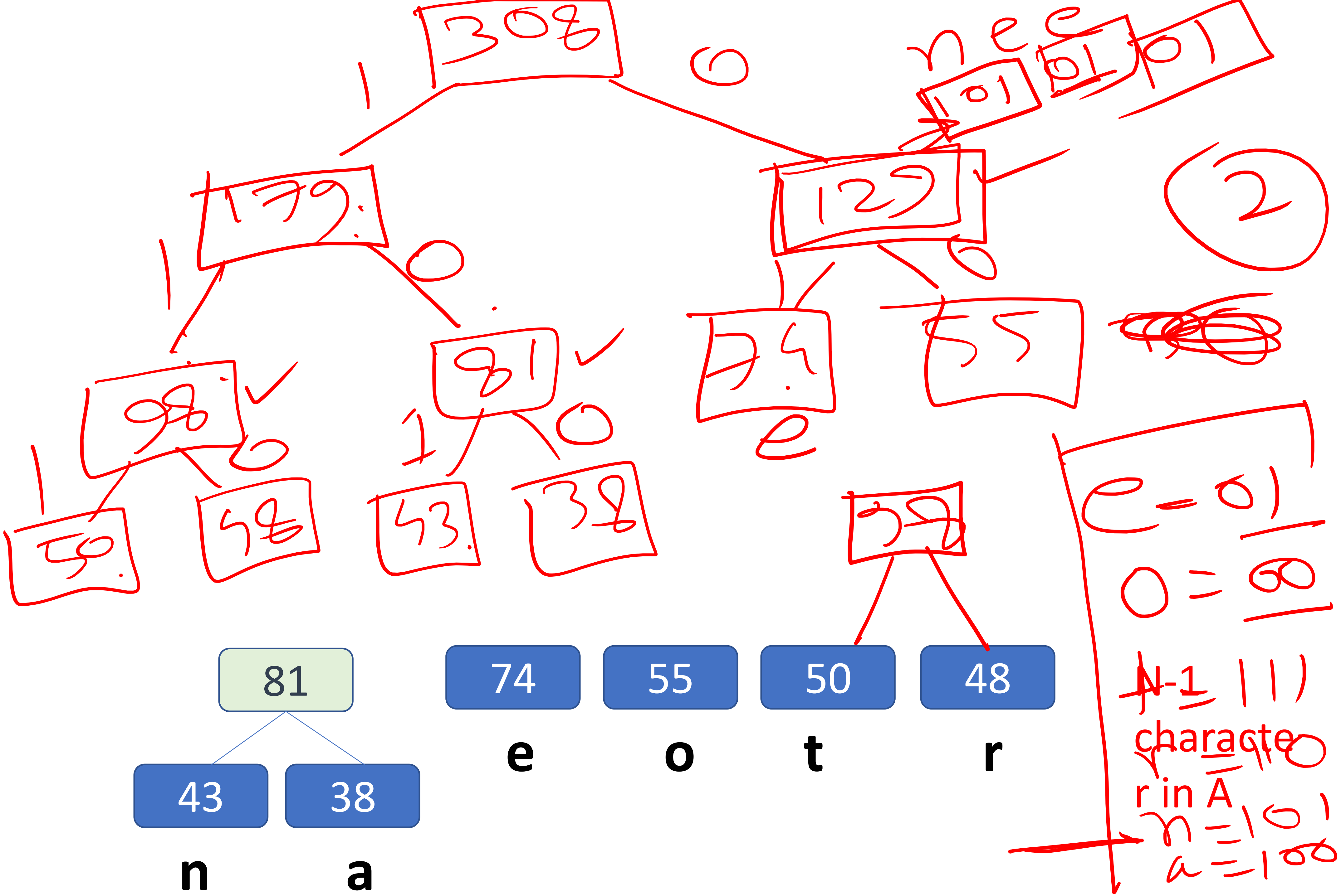
Cost of an encoding

$c \in A$	f_c	T	l_c
e:	**74**	000	3
o:	**55**	001	3
t:	**50**	010	3
r:	**48**	011	3
n:	**43**	100	3
a:	**38**	101	3
	308		**3**

$$B(T,\{f_C\}) = \sum_{c \in A} f_c l_c$$

924

c ∈ A	f_c
e:	74
o:	55
t:	50
r:	48
n:	43
a:	38



Objective

Given $\{f_c\}$, we can compute a prefix-free code using the Huffman's algorithm.

Prove the resulting tree is optimal one.

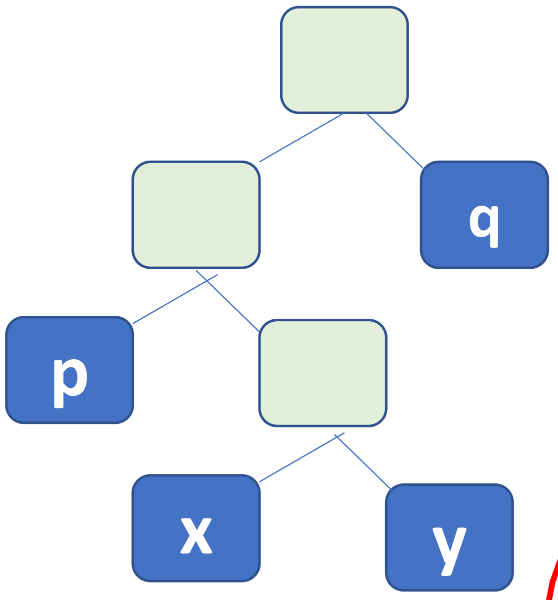
f_x f_y

Exchange Lemma

① Tree is Full

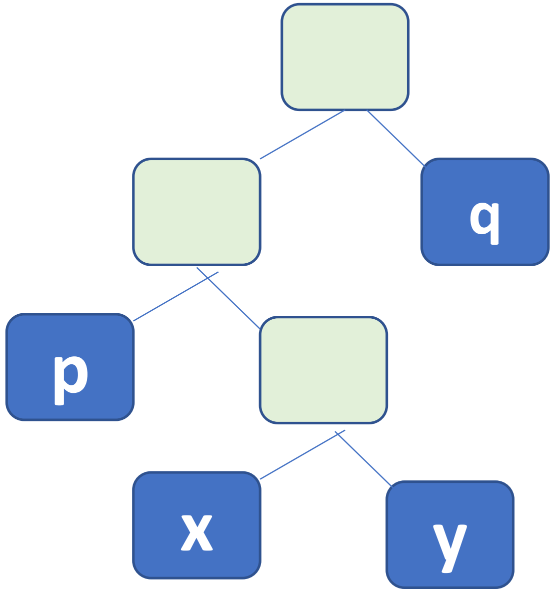
LEMMA:

Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



② x,y will be in the lowest level

LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.

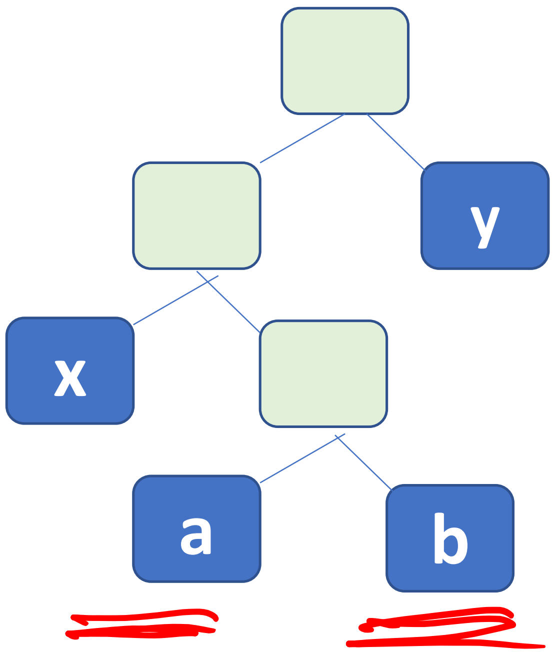


Let T be an optimal code (tree) for A alphabets.

CASE 1:

x,y are siblings in the tree, lemma holds

LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



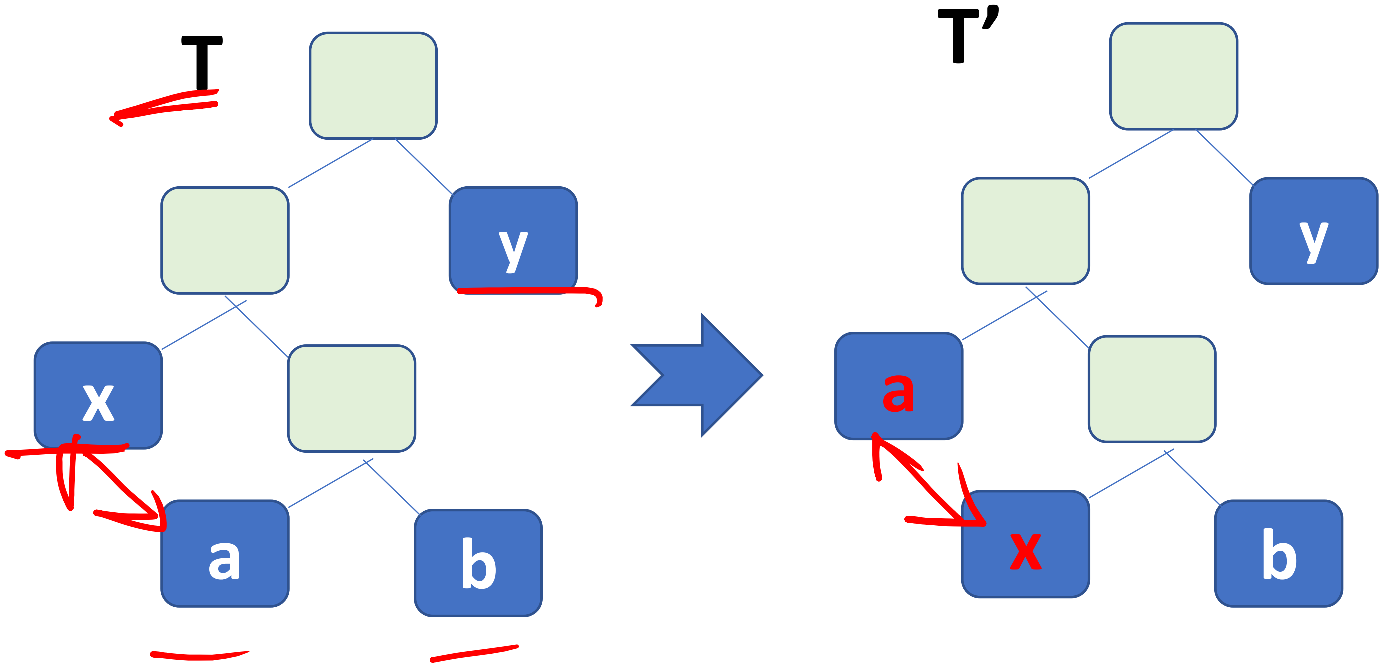
Let T be an optimal code (tree) for A alphabets.

CASE 2:
a, b are siblings in the lowest level of tree

Note: why a, b exist???

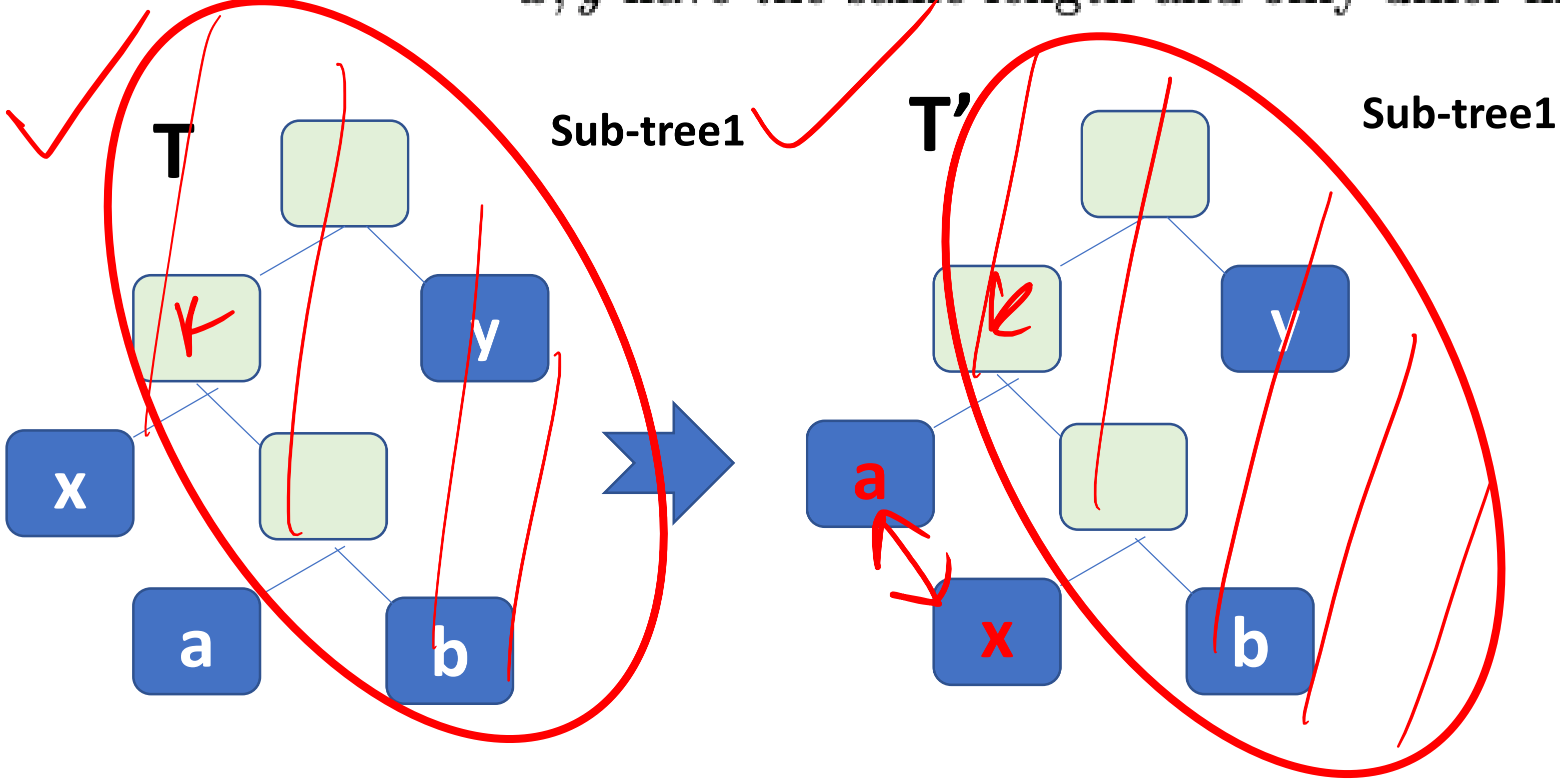
Because tree is full

LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



fx , fy have the smallest frequencies
And fx ≤ fa. And fy ≤ fb

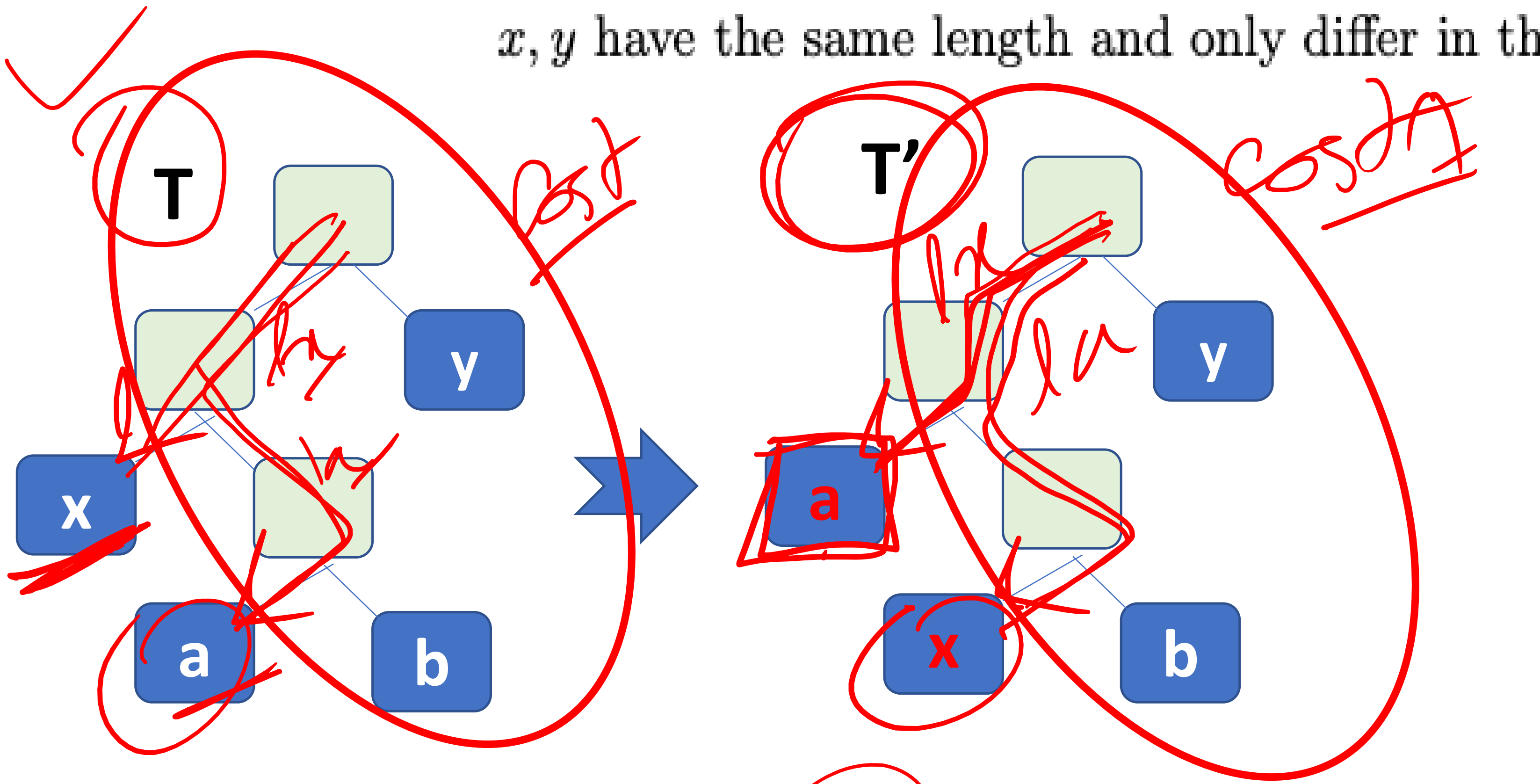
LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



fx , fy have the smallest frequencies

And fx ≤ fa. And fy ≤ fb

LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.

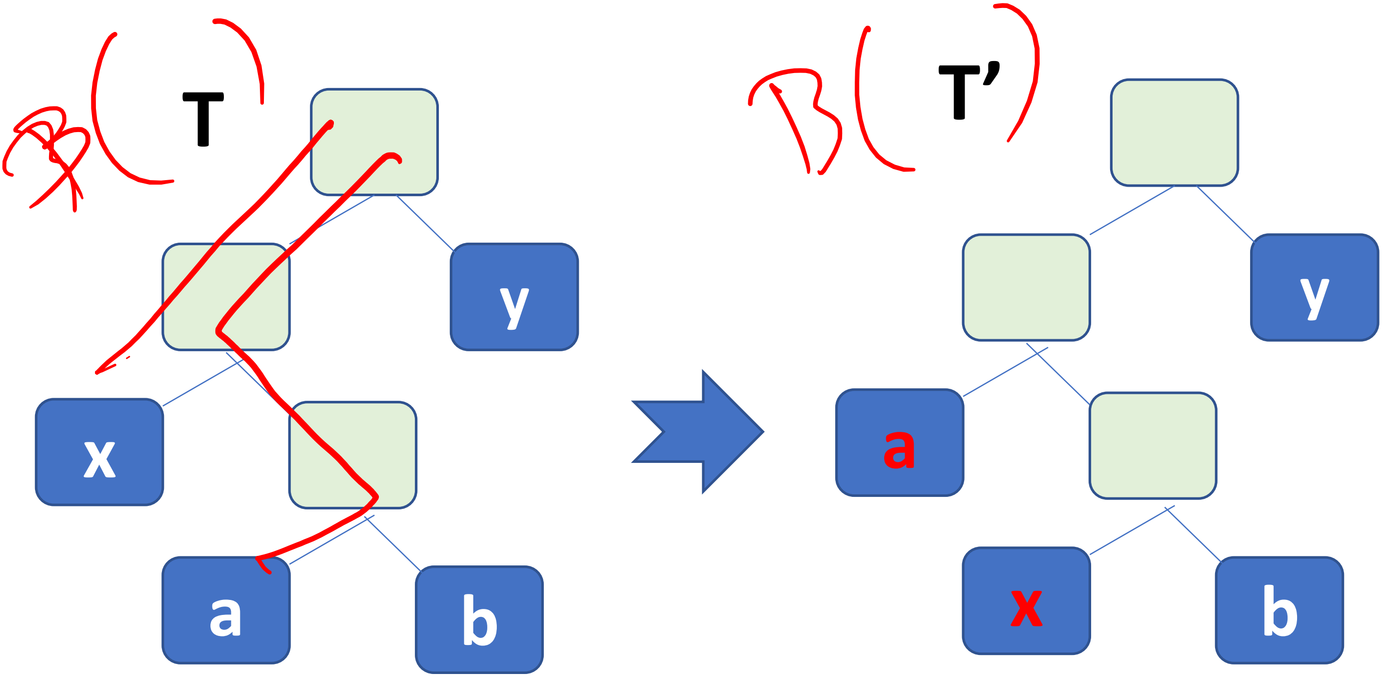


$B(T) = \text{Cost1} + f_x l_x + f_a l_a$ $B(T') = \text{Cost1} + f_x l_a + f_a l_x$

fx , fy have the smallest frequencies

And fx ≤ fa. And fy ≤ fb

LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



$$\boxed{B(T)-B(T')} \geq 0$$

$$= f_x l_x + f_a l_a - f_x l_a - f_a l_x$$

$$=$$

$$=$$

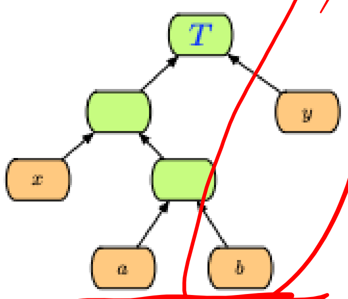
$$= \underbrace{(l_a - l_x)}_{\geq 0}\underbrace{(f_a - f_x)}_{\geq 0} \ \boxed{\geq 0}$$

$B(T) = \cancel{Cost1} + f_x l_x + f_a l_a$ $\quad$ $B(T') = \cancel{Cost1} + f_x l_a + f_a l_x$

fx , fy have the smallest frequencies

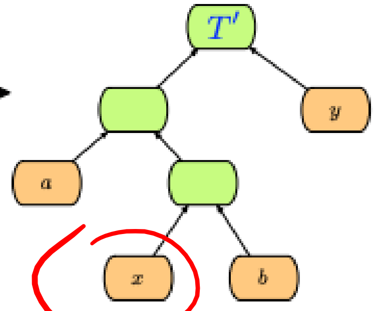
And fx ≤ fa. And fy ≤ fb

T



$f_x \leq f_a$

T'



But T is optimal!!

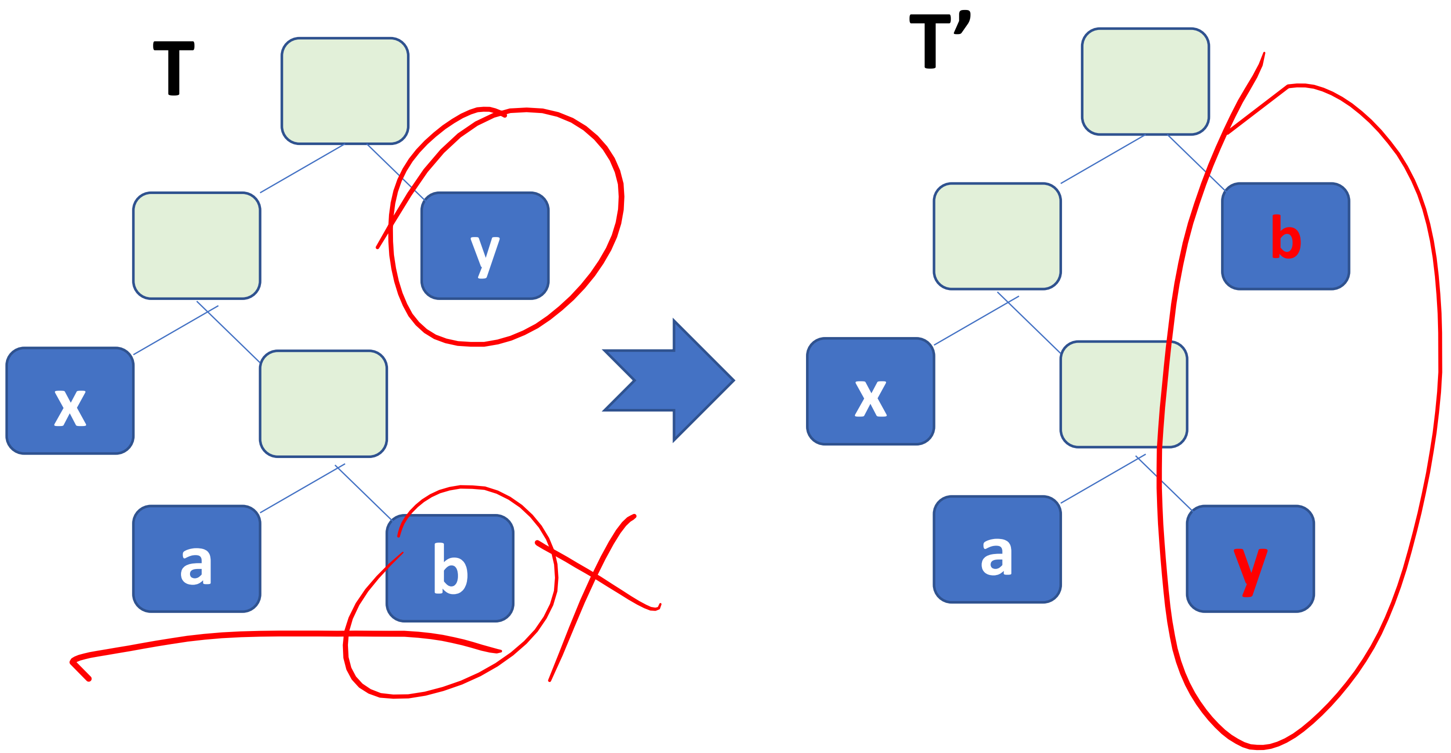
$$B(T) = \sum_c f_c \ell_c + f_x \ell_x + f_a \ell_a \qquad B(T') = \sum_c f_c \ell'_c + f_x \ell'_x + f_a \ell'_a$$

T optimal

$$B(T) - B(T') \geq 0$$

LEMMA:

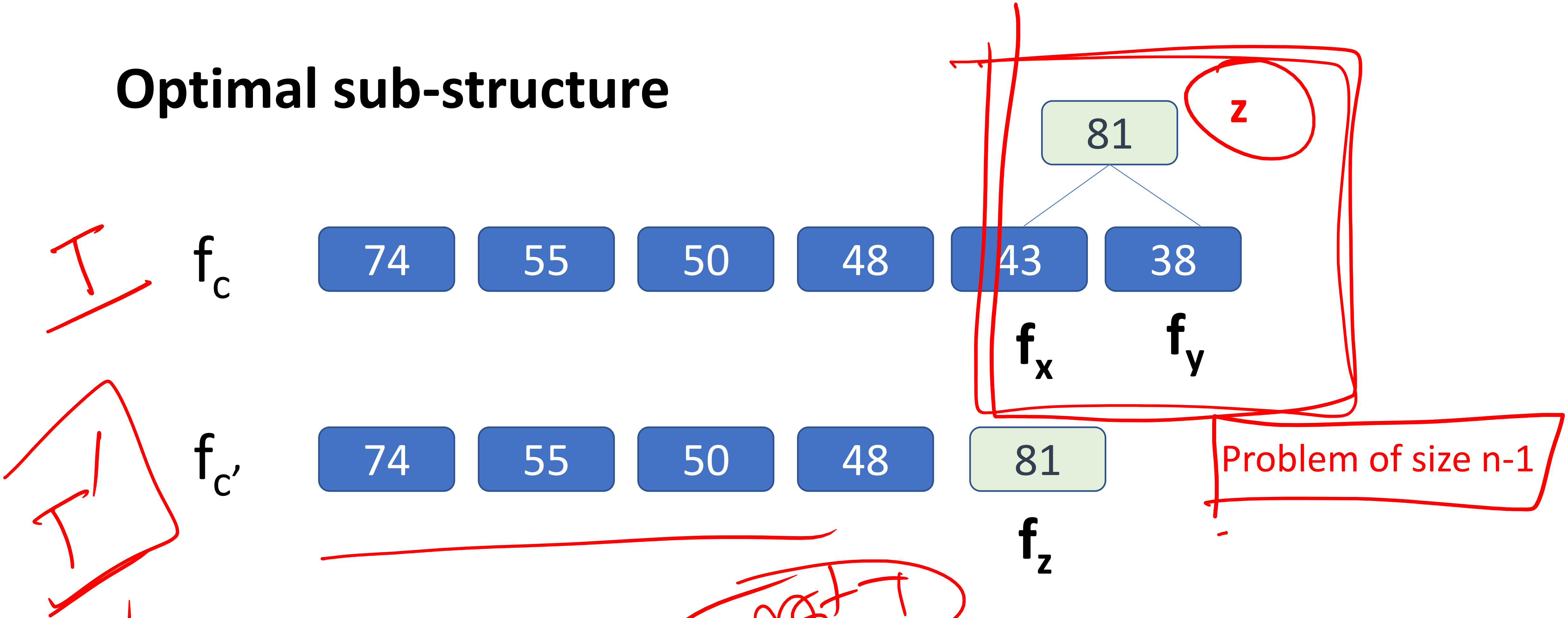
Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for ***A*** in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



Do the same argument with b and y

fx , fy have the smallest frequencies
And fx ≤ fa. And fy ≤ fb

Optimal sub-structure

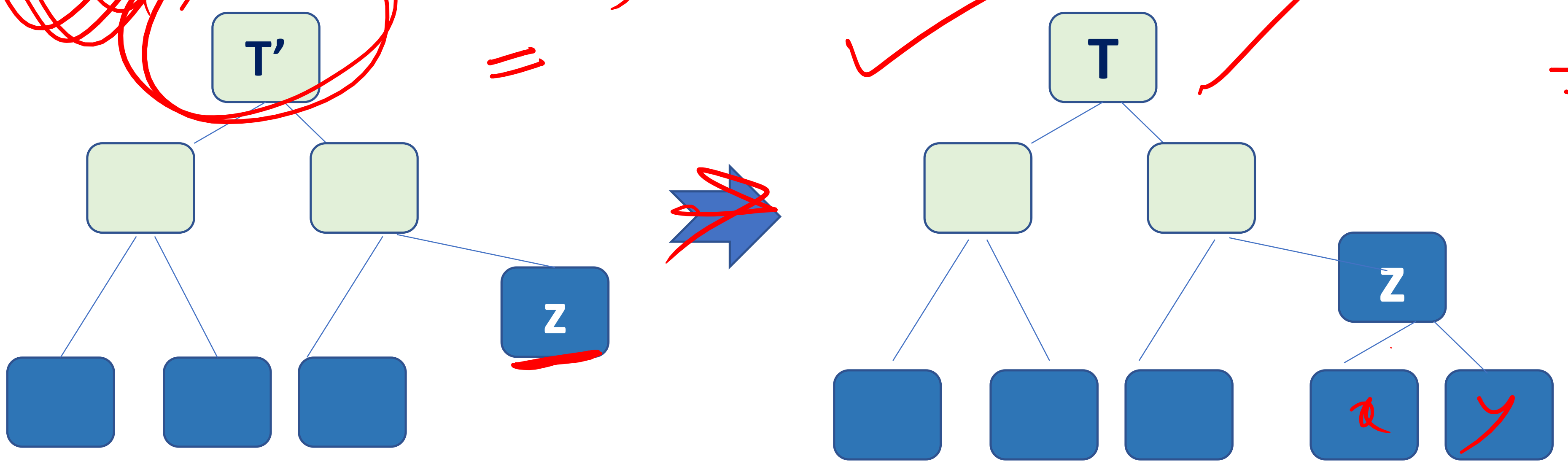


Lemma: the optimal solution for T consists of computing an optimal solution for T' and replacing the left z with a node having children x,y

optimal

$B(T') - B(T) =$

$B(T) - B(T') = \boxed{f_x + f_y}$



$$\boxed{B(T') = B(T) - f_x - f_y}$$

$l_x = l_z + 1$

$l_y = l_z + 1$

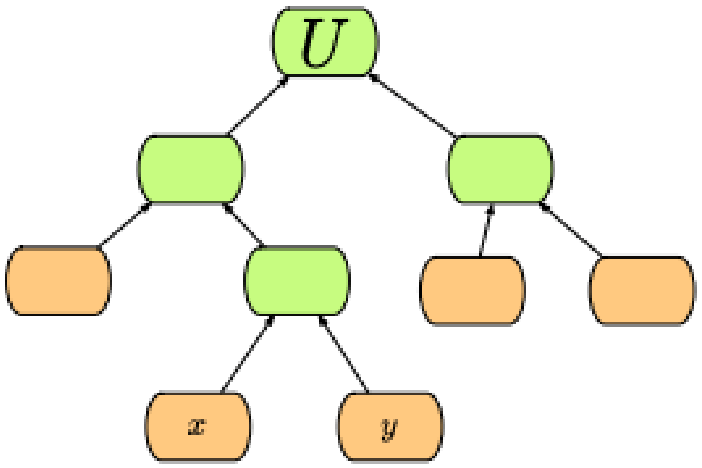
$f_z = f_x + f_y$

$B(T') = (f_x + f_y)(l_z)$

$B(T) = f_x(l_z + 1) + f_y(l_z + 1)$

Suppose T is not optimal

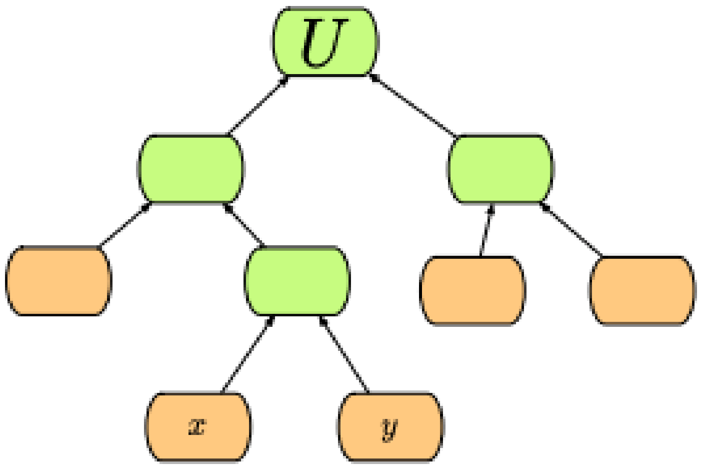
There exist another tree U such that $B(U) < B(T)$



X and y are siblings in U

Suppose T is not optimal

There exist another tree U such that B(U) < B(T)



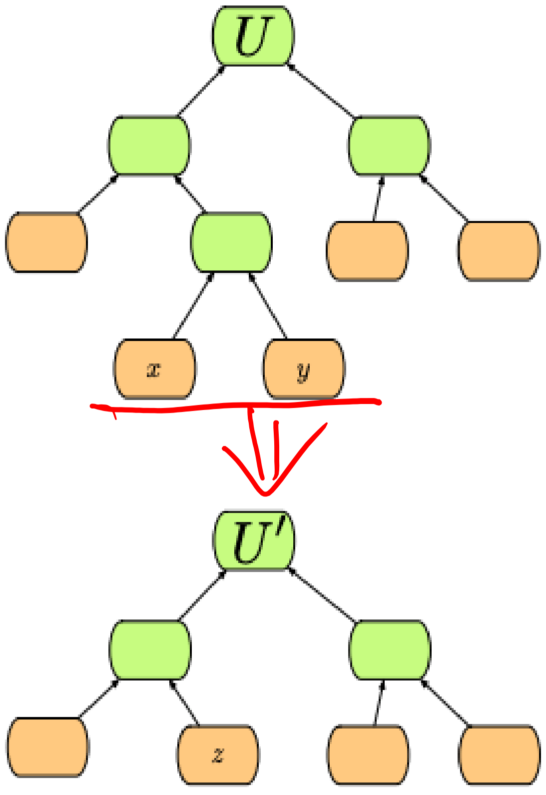
X and y are siblings in U

B(T')

Suppose T is not optimal

There exist another tree U such that $\boxed{B(U) < B(T)}$

U'



Define a tree U' such that x,y are combined to z

$$B(U') = B(U) - f_x - f_y$$

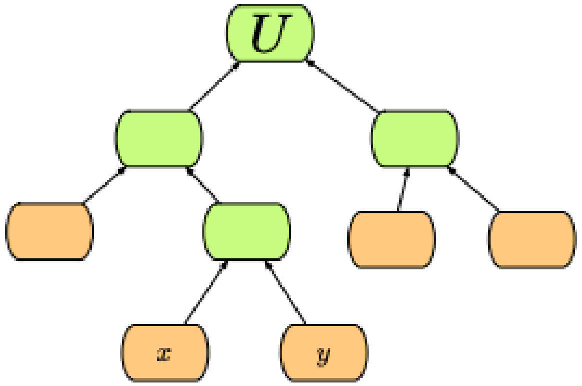
$$< B(T) - f_x - f_y \quad = B(T')$$

$$= B(T')$$

$$B(U') < B(T')$$

Suppose T is not optimal

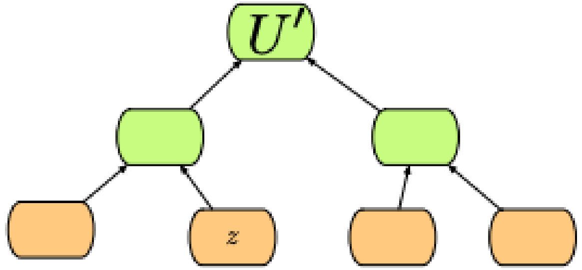
Conflicts with the assumption that T' is not optimal. Hence, T must have also been optimal.



$$B(U) < B(T)$$

$$B(U') = B(U) - f_x - f_y$$

$< B(t)$ - fx - fy



But this implies that B(T') was not optir