L11

4800

10.18.2016

abhi shelat

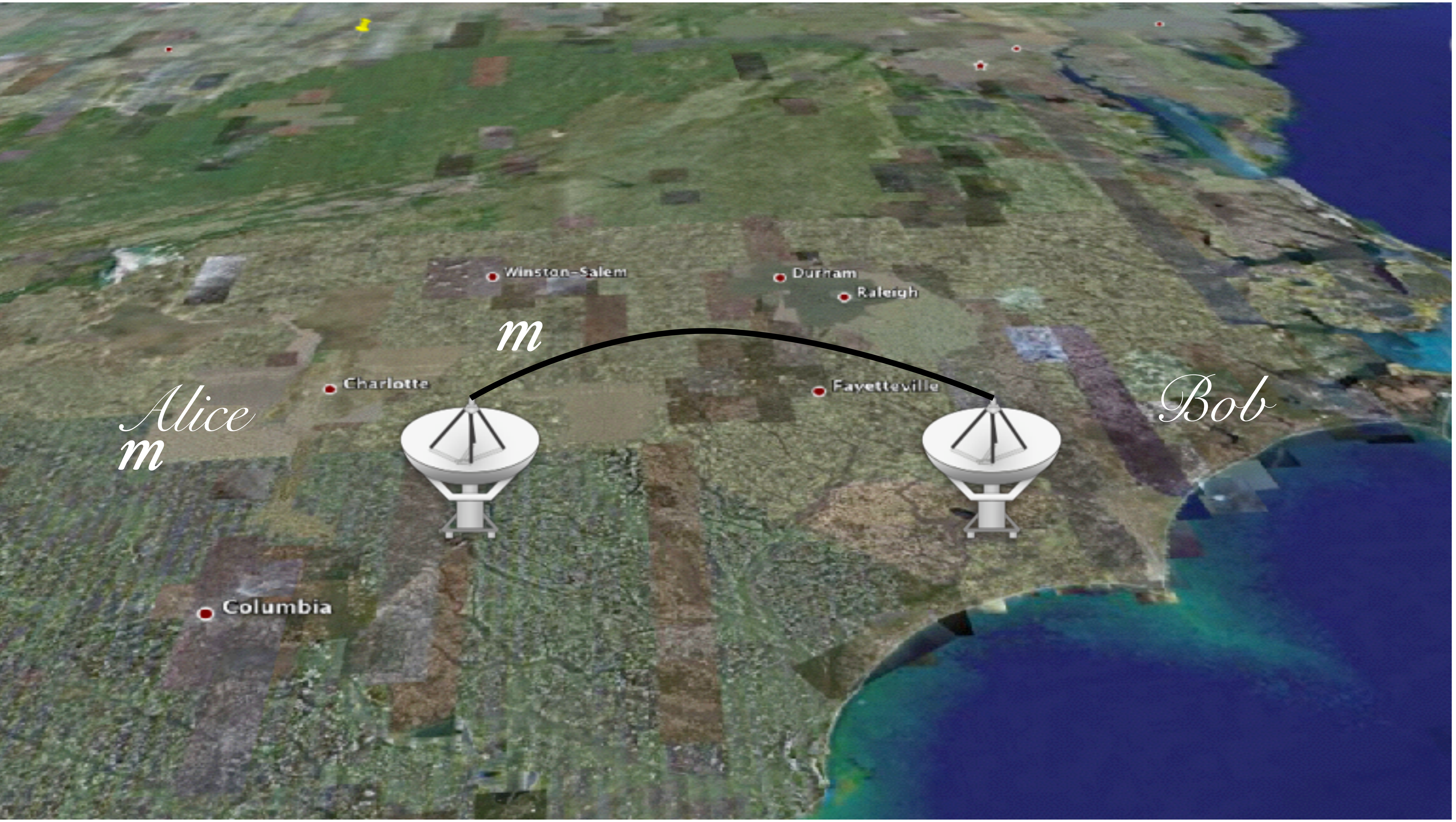
Huffman

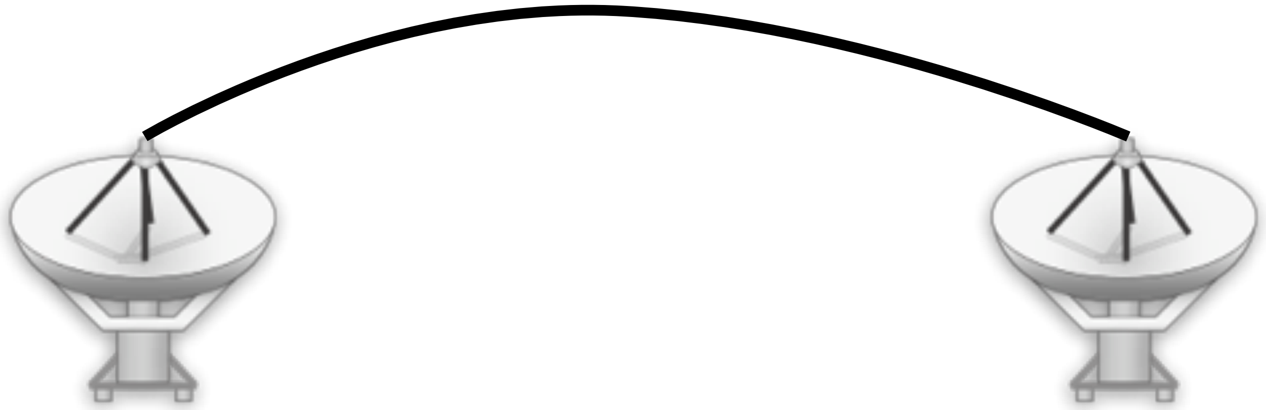
CS4800

abhi shelat


image: wikimedia

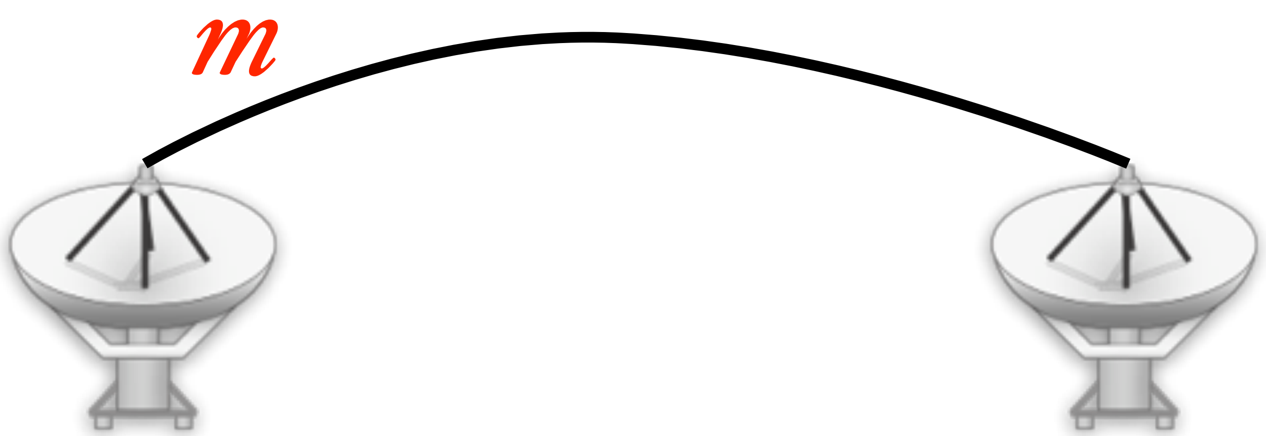

Winston-Salem
Durham
Raleigh
Charlotte
Fayetteville
Alice
m
Bob
Columbia


Winston-Salem
Durham
Raleigh
m
Charlotte
Fayetteville
Alice
m
Bob
Columbia



MOSCOW — President Vladimir V. Putin’s typically theatrical order to withdraw the bulk of Russian forces from Syria, a process that the Defense Ministry said it began on Tuesday, seemingly caught Washington, Damascus and everybody in between off guard — just the way the Russian leader likes it.

By all accounts, Mr. Putin delights at creating surprises, reinforcing Russia’s newfound image as a sovereign, global heavyweight and keeping him at the center of world events



MOSCOW — President Vladimir V. Putin’s typically theatrical order to withdraw the bulk of Russian forces from Syria, a process that the Defense Ministry said it began on Tuesday, seemingly caught Washington, Damascus and everybody in between off guard — just the way the Russian leader likes it.

By all accounts, Mr. Putin delights at creating surprises, reinforcing Russia’s newfound image as a sovereign, global heavyweight and keeping him at the center of world events

$c \in C$ f_c T

e: 235
i: 200
o: 170
u: 87
p: 78
g: 47
b: 40
f: 24

881

$c \in C$	f_c	T	ℓ_c
e:	235	000	3
i:	200	001	3
o:	170	010	3
u:	87	011	3
p:	78	100	3
g:	47	101	3
b:	40	110	3
f:	24	111	3

881

→ 2643

def: cost of an encoding

code

frequencies of your messages

length of code for character c in code T

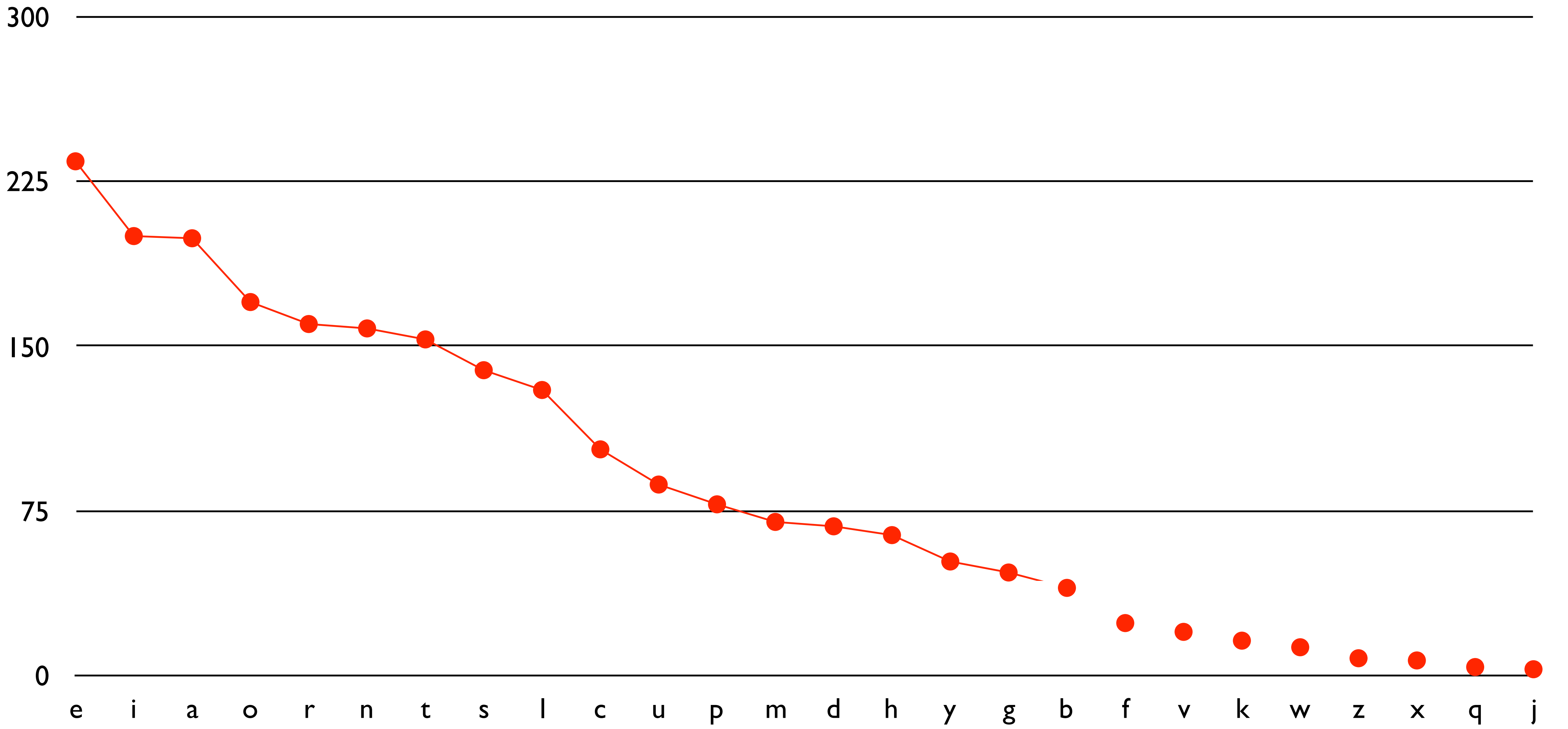
$$\underline{B(T,\{f_c\})} = \sum_{\underline{c\in C}} \underbrace{f_c}_{} \cdot \ell_c$$

$c \in C$	f_c	T	ℓ_c
e:	235	000	3
i:	200	001	3
o:	170	010	3
u:	87	011	3
p:	78	100	3
g:	47	101	3
b:	40	110	3
f:	24	111	3
	881		

$\cdot 3 = 2643$

character frequency

```
e: 234803
i: 200613
a: 198938
o: 170392
r: 160491
n: 158281
t: 152570
s: 139238
l: 130172
c: 103307
 u: 87211
 p: 78077
 m: 70504
 d: 68007
 h: 64165
 y: 51527
 g: 47011
 b: 40351
 f: 24110
 v: 20103
 k: 16012
 w: 13825
  z: 8439
  x: 6926
  q: 3729
  j: 3075
```



morse code

International Morse Code

- 1 dash = 3 dots.
- The space between parts of the same letter = 1 dot.
- The space between letters = 3 dots.
- The space between words = 7 dots.

A	• ▬	V	• • • ▬
B	▬ • • •	W	• ▬ ▬
C	▬ • ▬ •	X	▬ • • ▬
D	▬ • •	Y	▬ • ▬ ▬
E	•	Z	▬ ▬ • •
F	• • ▬ •	.	• ▬ • ▬ • ▬
G	▬ ▬ •	,	▬ ▬ • • ▬ ▬
H	• • • •	?	• • ▬ ▬ • •
I	• •	/	▬ • • ▬ •
J	• ▬ ▬ ▬	@	• ▬ ▬ • ▬ •
K	▬ • ▬	1	• ▬ ▬ ▬ ▬
L	• ▬ • •	2	• • ▬ ▬ ▬
M	▬ ▬	3	• • • ▬ ▬
N	▬ •	4	• • • • ▬
O	▬ ▬ ▬	5	• • • • •
P	• ▬ ▬ •	6	▬ • • • •
Q	▬ ▬ • ▬	7	▬ ▬ • • •
R	• ▬ •	8	▬ ▬ ▬ • •
S	• • •	9	▬ ▬ ▬ ▬ •
T	▬	0	▬ ▬ ▬ ▬ ▬
U	• • ▬		

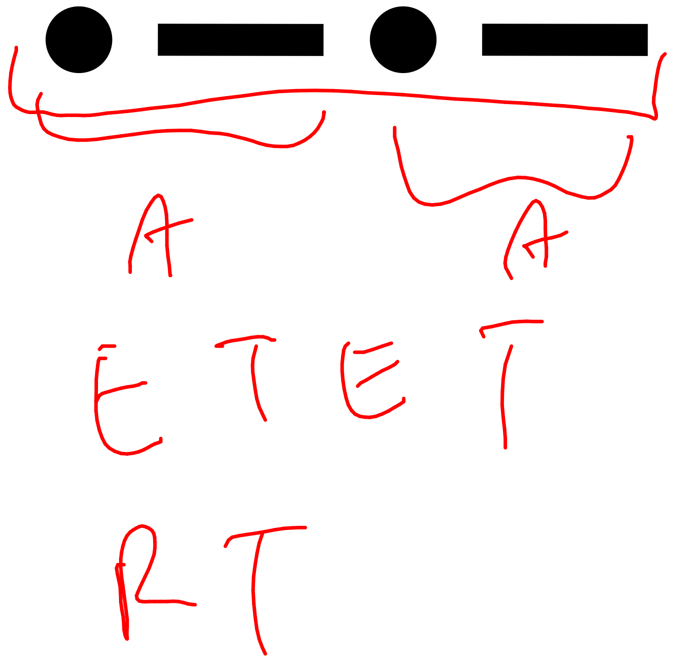
image http://en.wikipedia.org/wiki/Morse_code

morse code

International Morse Code

- 1 dash = 3 dots.
- The space between parts of the same letter = 1 dot.
- The space between letters = 3 dots.
- The space between words = 7 dots.

A	● ▬	V	● ● ● ▬
B	▬ ● ● ●	W	● ▬ ▬
C	▬ ● ▬ ●	X	▬ ● ● ▬
D	▬ ● ●	Y	▬ ● ▬ ▬
E	●	Z	▬ ▬ ● ●
F	● ● ▬ ●	.	● ▬ ● ▬ ● ▬
G	▬ ▬ ●	,	▬ ▬ ● ● ▬ ▬
H	● ● ● ●	?	● ● ▬ ▬ ● ●
I	● ●	/	▬ ● ● ▬ ●
J	● ▬ ▬ ▬	@	● ▬ ▬ ● ▬ ●
K	▬ ● ▬	1	● ▬ ▬ ▬ ▬
L	● ▬ ● ●	2	● ● ▬ ▬ ▬
M	▬ ▬	3	● ● ● ▬ ▬
N	▬ ●	4	● ● ● ● ▬
O	▬ ▬ ▬	5	● ● ● ● ●
P	● ▬ ▬ ●	6	▬ ● ● ● ●
Q	▬ ▬ ● ▬	7	▬ ▬ ● ● ●
R	● ▬ ●	8	▬ ▬ ▬ ● ●
S	● ● ●	9	▬ ▬ ▬ ▬ ●
T	▬	0	▬ ▬ ▬ ▬ ▬
U	● ● ▬		



def: prefix-free code

Code such that for any two symbols $x, y \in C$ $x \neq y$, code(x) is not a prefix of code(y)

def: prefix-free code

$$\forall x, y \in C, x \neq y \implies \textsc{code}(x) \text{ not a prefix of } \textsc{code}(y)$$

def: prefix code

$$\forall x, y \in C, x \neq y \implies \textsc{code}(x) \text{ not a prefix of } \textsc{code}(y)$$

e:	235	0
i:	200	10
o:	170	110
u:	87	1110
p:	78	11110
g:	47	111110
b:	40	1111110
f:	24	11111110

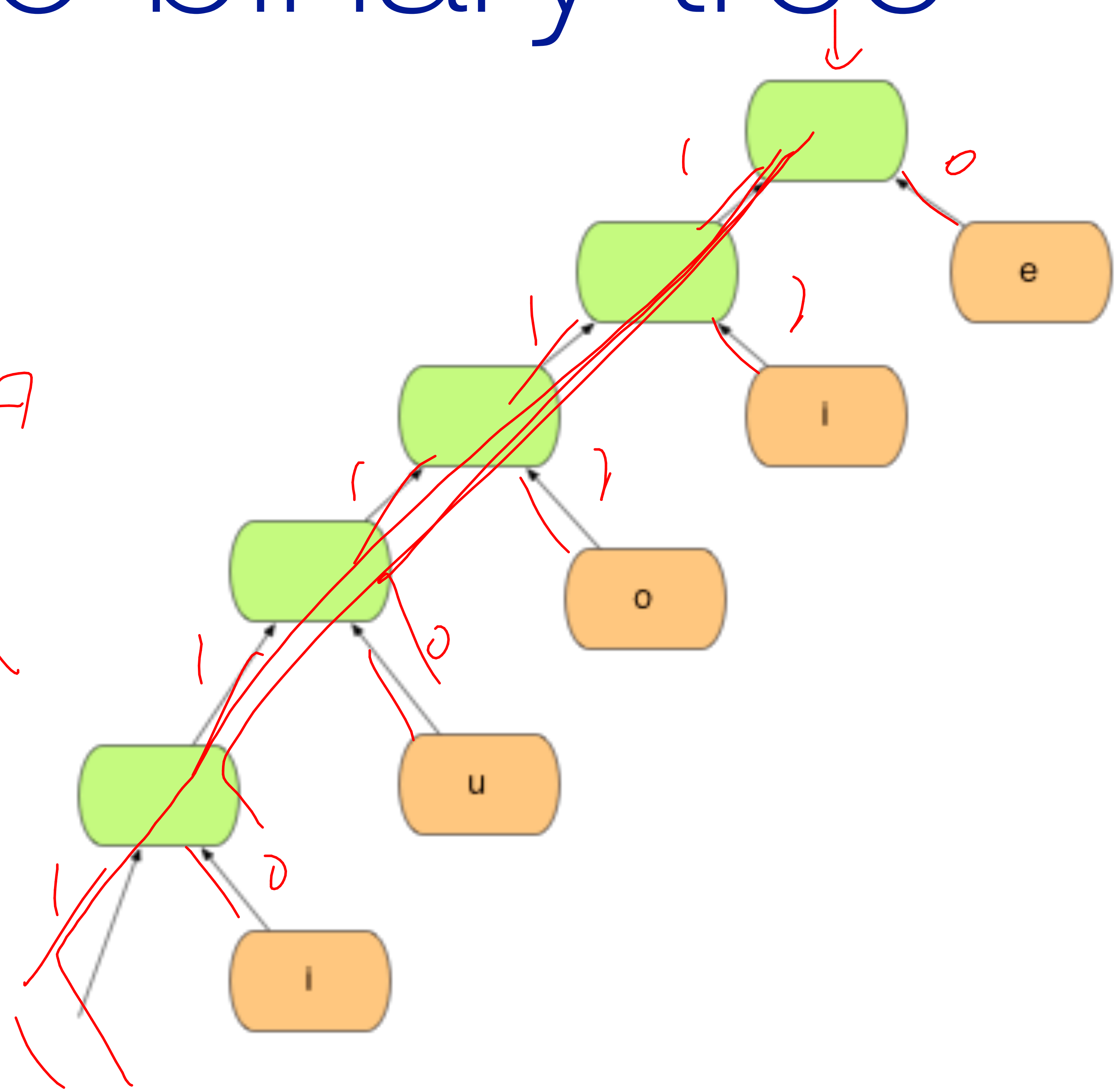
decoding a prefix code

e:	235	0
i:	200	10
o:	170	110
u:	87	1110
p:	78	11110
g:	47	111110
b:	40	1111110
f:	24	11111110

111111010111110

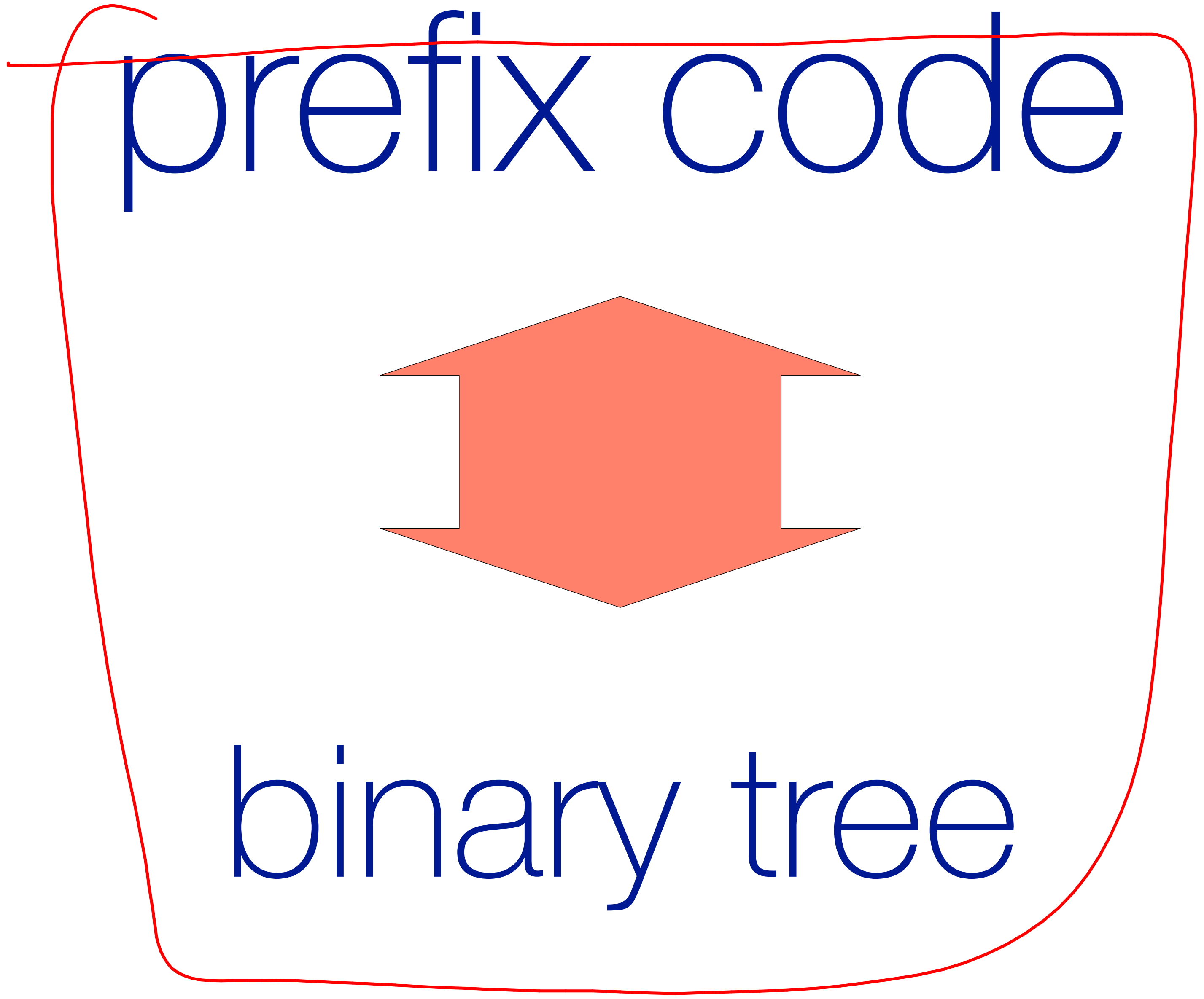
code to binary tree

e: 235	0
i: 200	10
o: 170	110
u: 87	1110
p: 78	11110
g: 47	111110
b: 40	1111110
f: 24	11111110



111111010111110

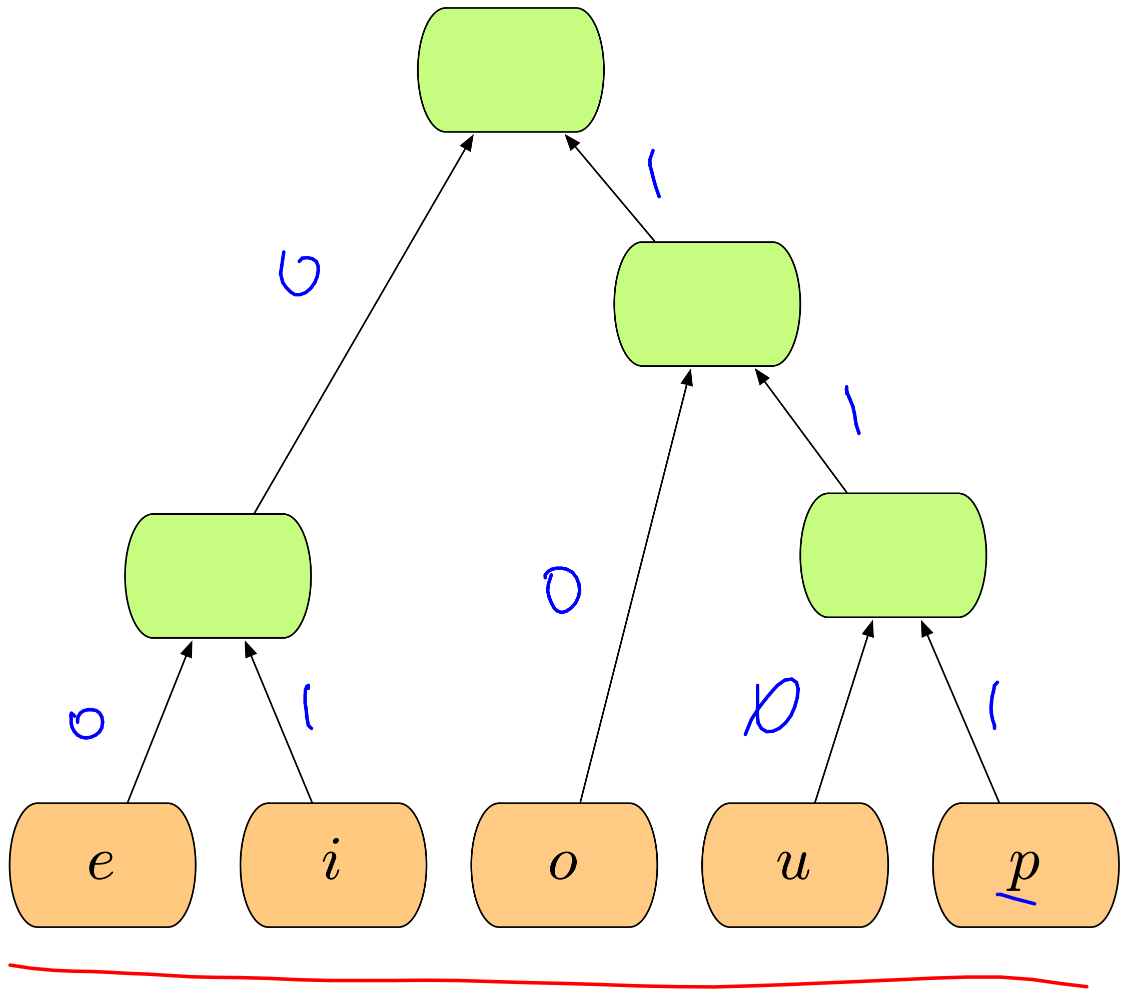
B I G



prefix code

binary tree

use tree to encode



$c \in C$	f_c	T	ℓ_c
e:	235	00	2
i:	200	01	2
o:	170	10	2
u:	87	110	3
p:	78	111	3

goal

given the frequencies for some message space $\underline{\{f_c\}}_{c\in C}$

all frequencies are > 0.

design an optimal prefix free code.

goal

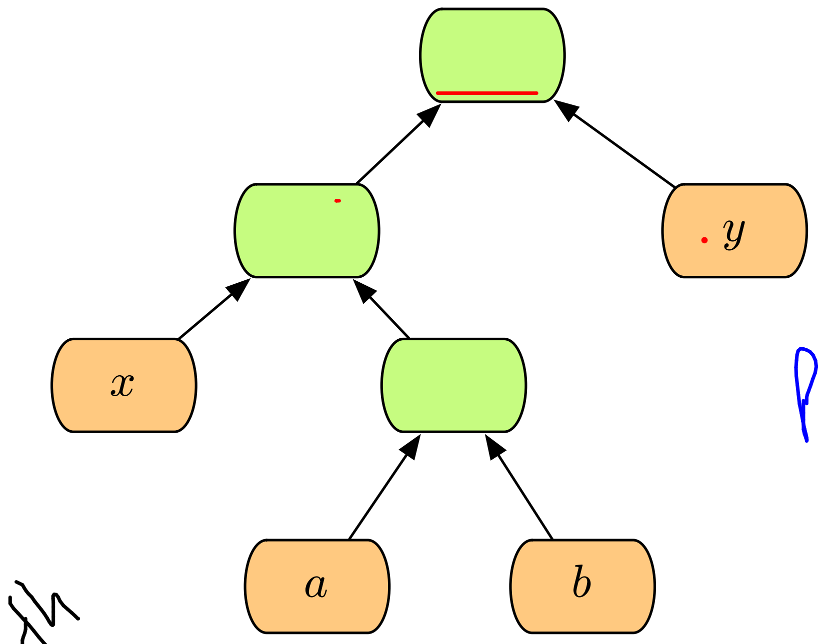
(all frequencies are > 0)

given the character frequencies $\underline{\{f_c\}}_{c \in C}$

produce a prefix code T with smallest cost

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

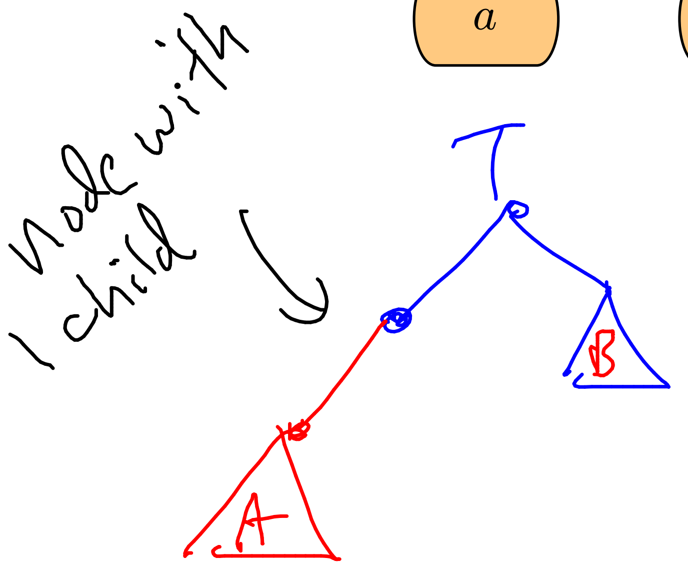
property

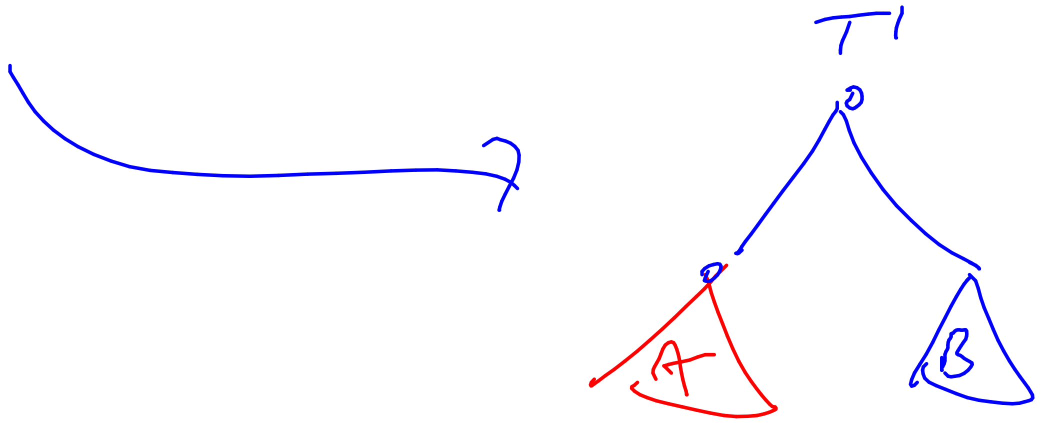


lemma:optimal tree must be full.

each node has either 0 children or 2 children

Prove: Suppose a code had a node with only 1 child. One could remove this node to produce a code with shorter codewords for several symbols

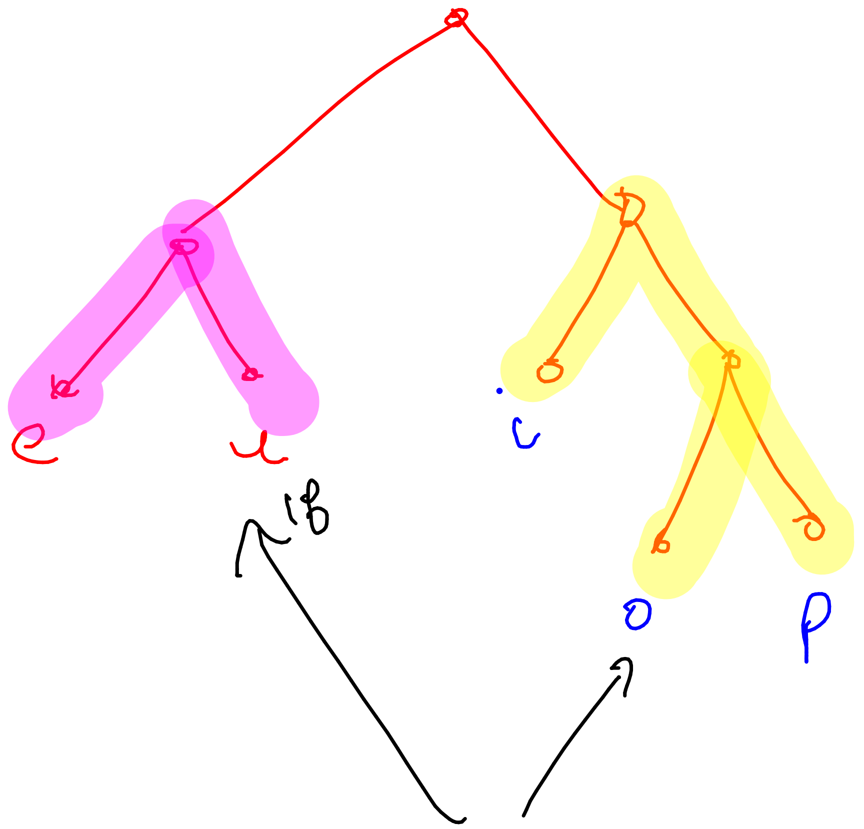




divide & conquer?

counter-example

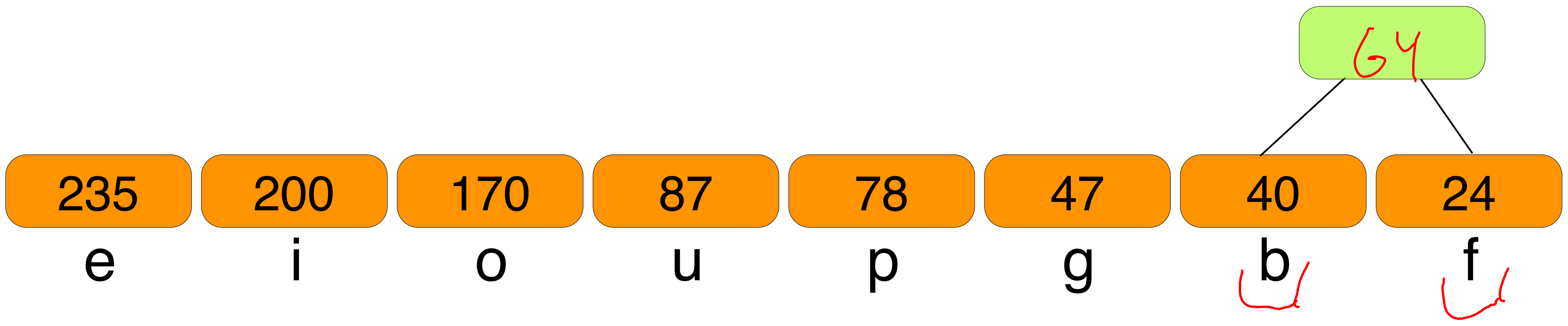
e: 32
i: 25
o: 20
u: 18
p: 5



e : 2
i : 2
o : 3
u : 2
p : 3

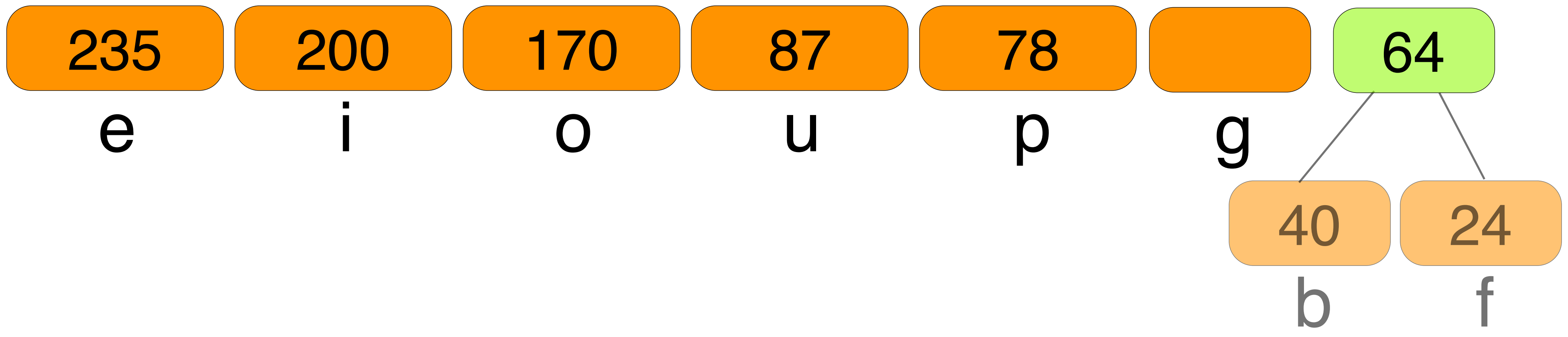
$18 \cdot 2 + 20 \cdot 3 = 96$

$20 \cdot 2 + 18 \cdot 3 = 94$



235	200	170	87	78	47	40	24
e	i	o	u	p	g	b	f

64

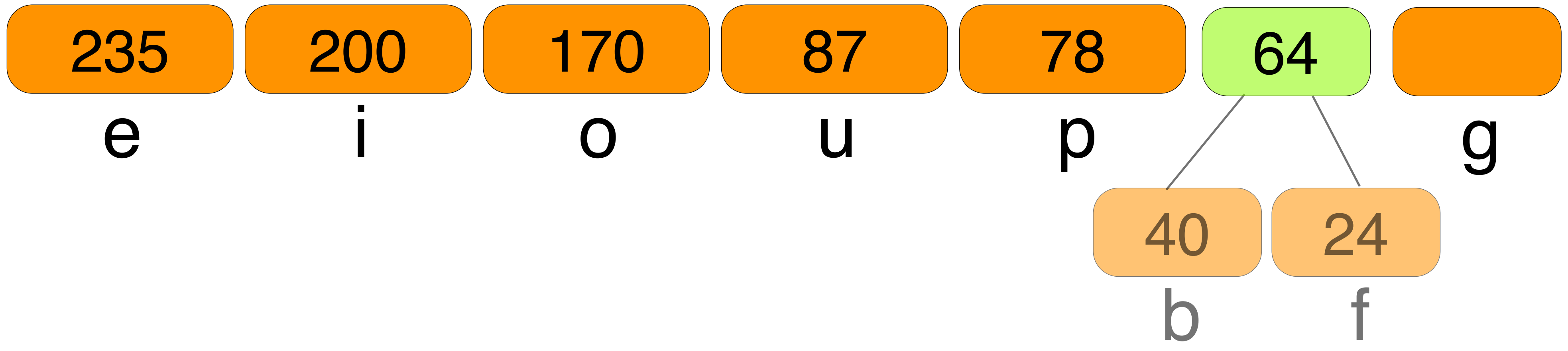


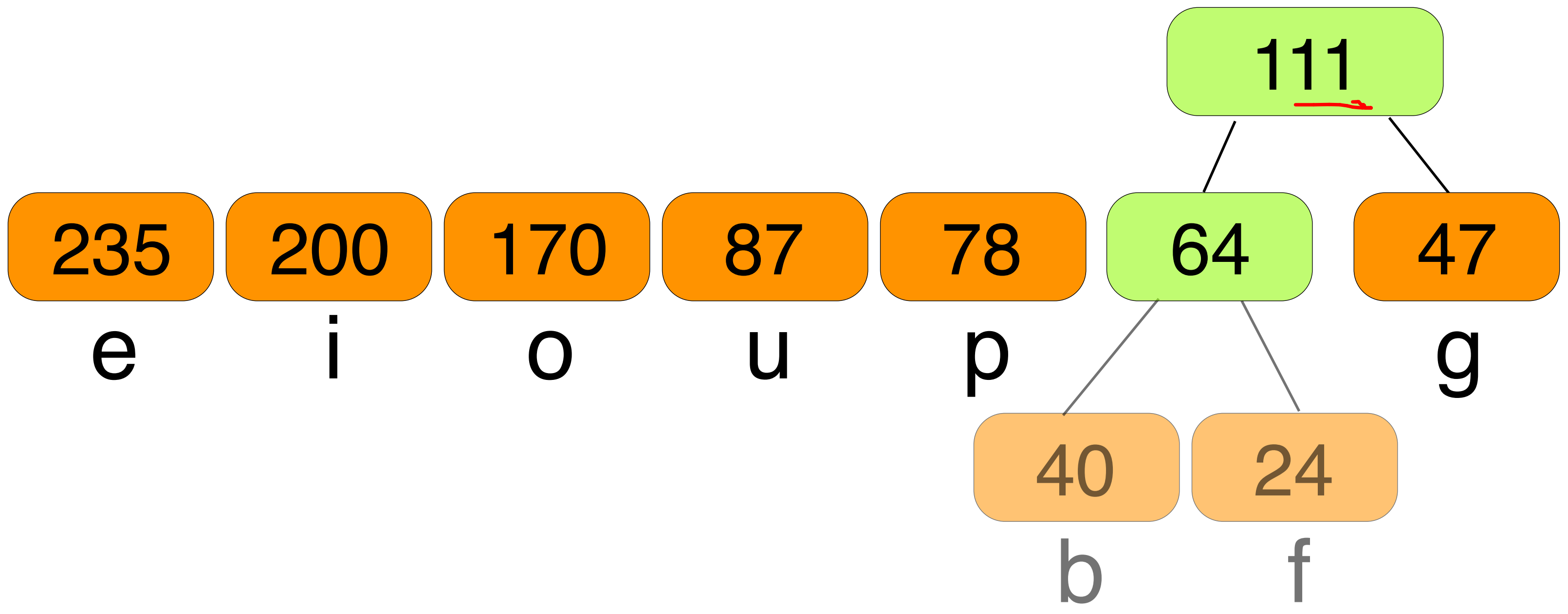
235 200 170 87 78 64

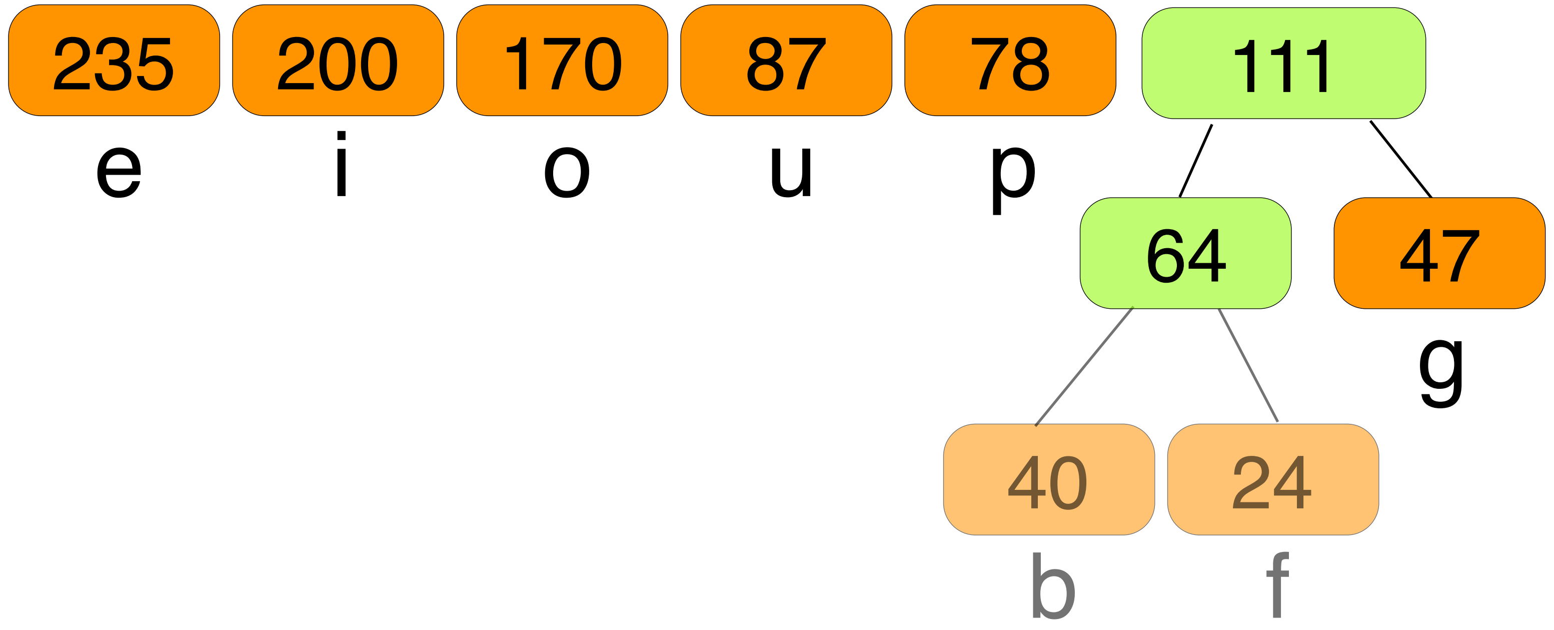
e i o u p g

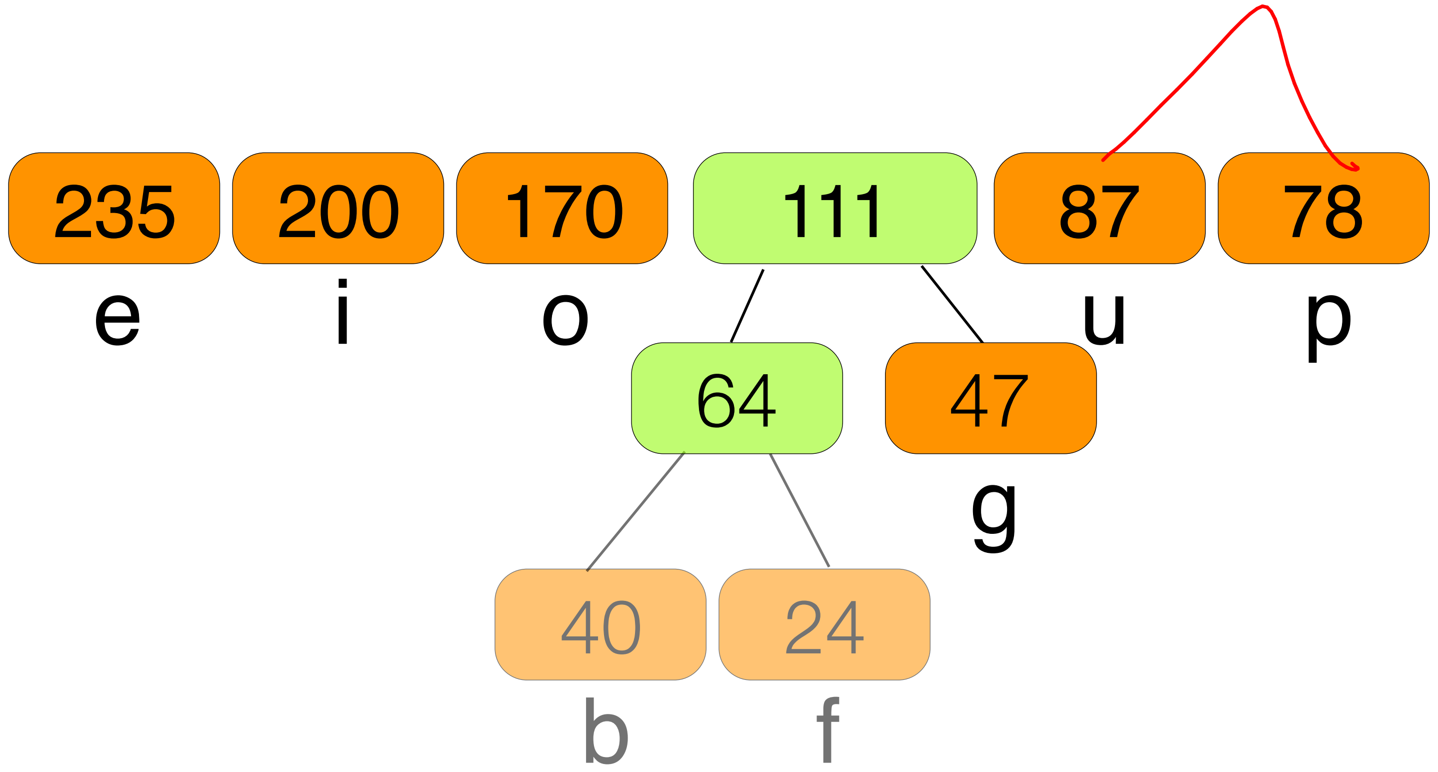
40 24

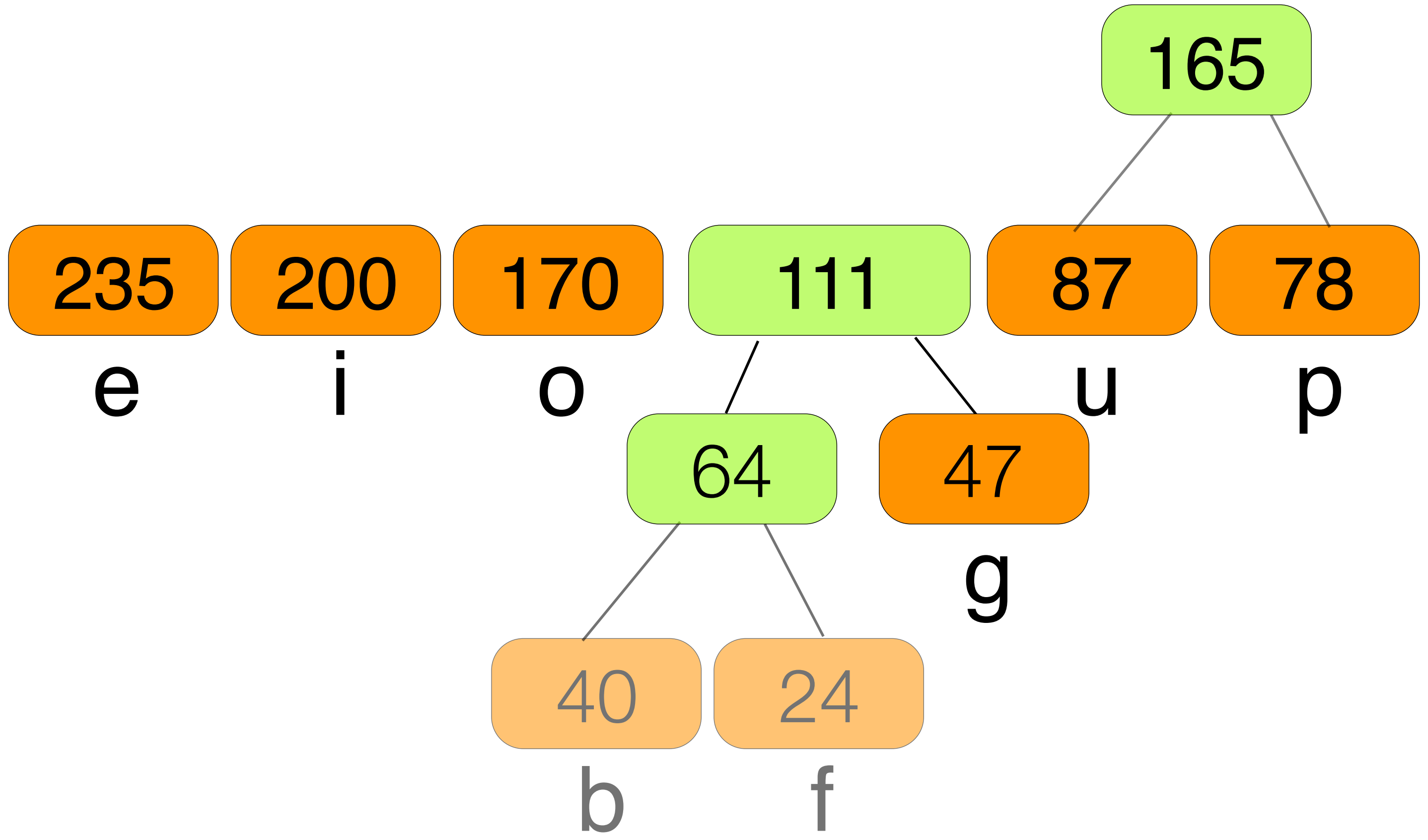
b f

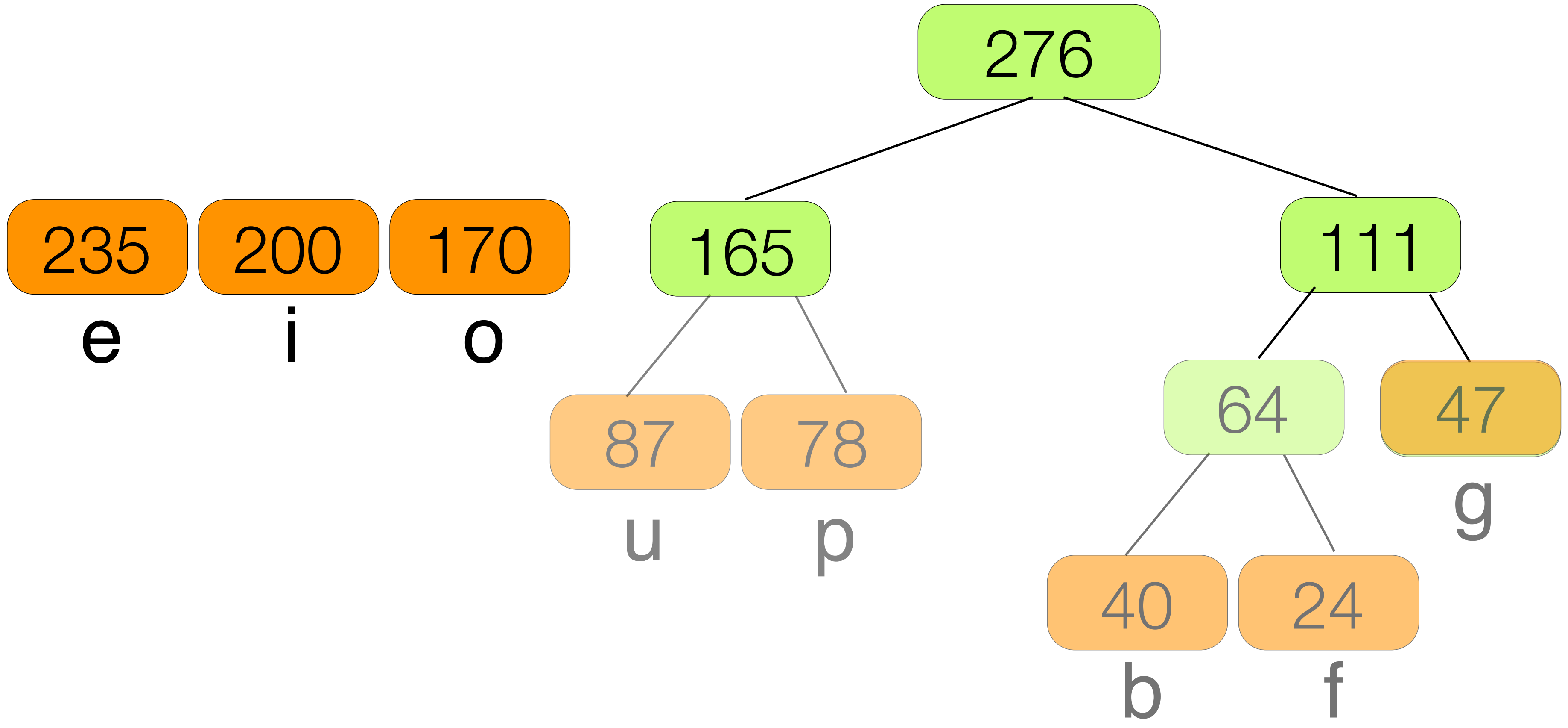

235
e
200
i
170
o
87
u
78
p
64
g
40
b
24
f

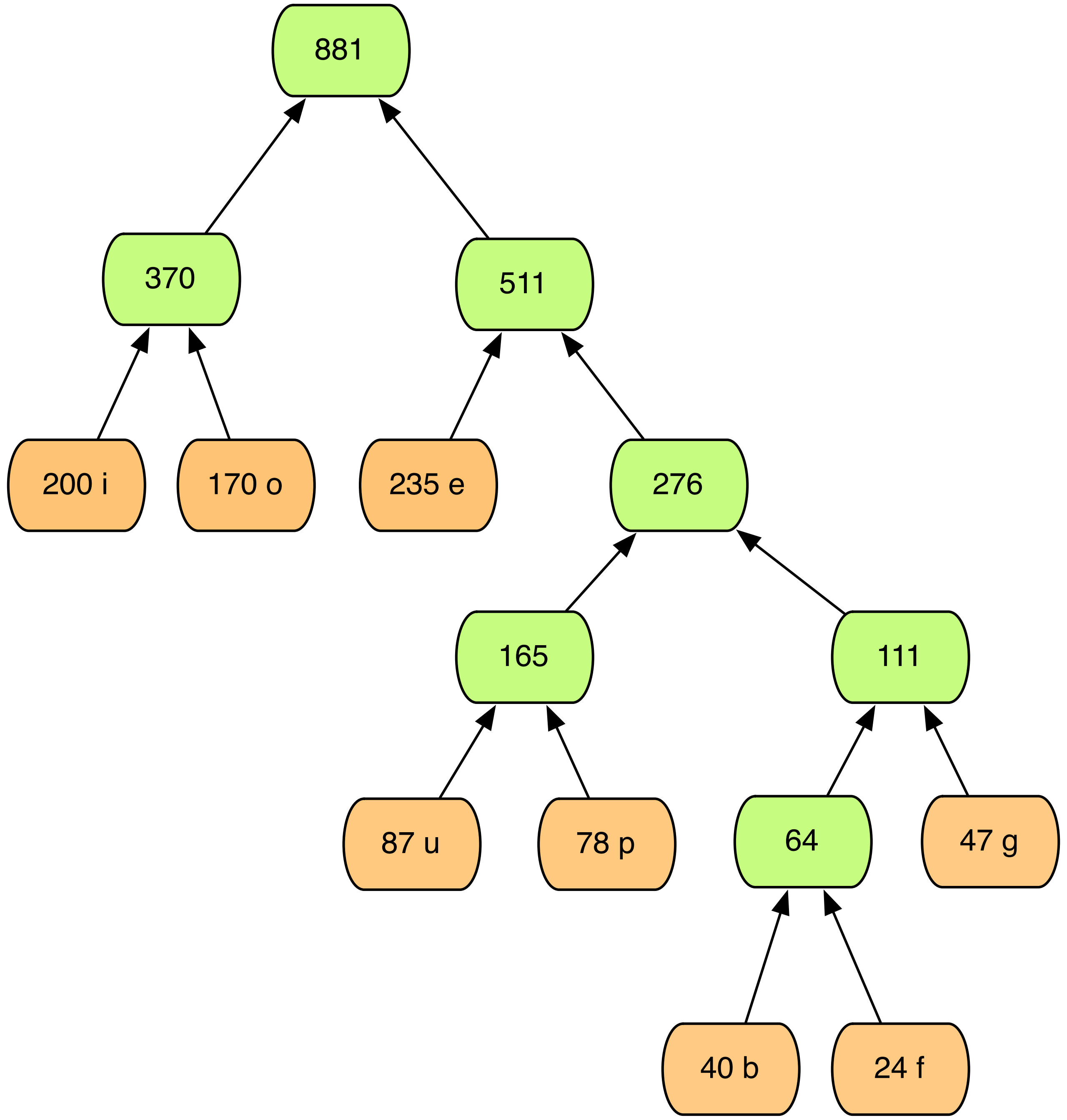

111
235
200
170
87
78
64
47
e
i
o
u
p
g
40
24
b
f

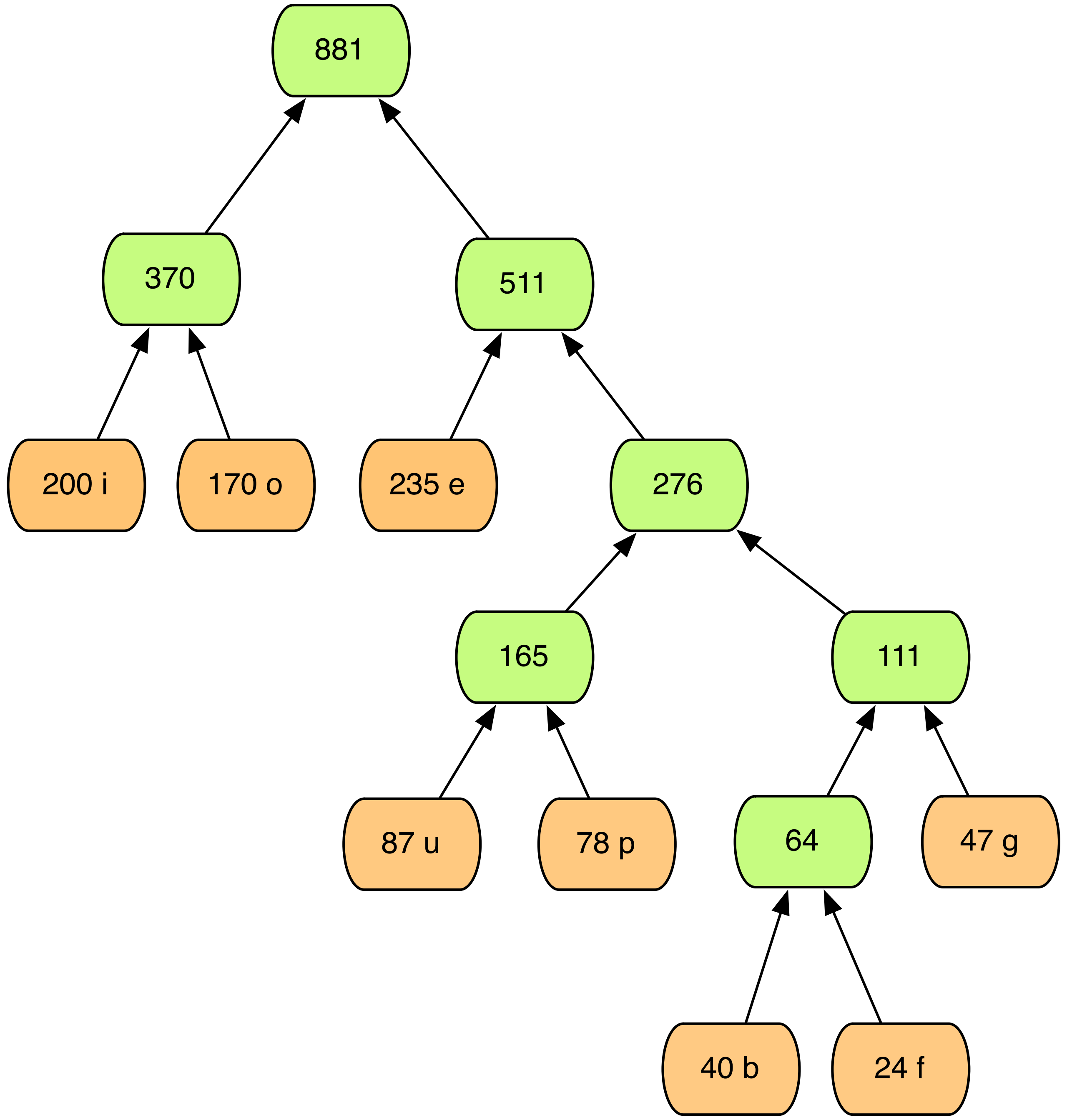

235
e
200
i
170
o
87
u
78
p
111
64
47
g
40
b
24
f


235
e
200
i
170
o
111
87
u
78
p
64
47
g
40
b
24
f


165
235
200
170
111
87
78
e
i
o
u
p
64
47
g
40
24
b
f


235
e
200
i
170
o
276
165
111
87
u
78
p
64
47
g
40
b
24
f


881
370
511
200 i
170 o
235 e
276
165
111
87 u
78 p
64
47 g
40 b
24 f



e: 235 01

i: 200 11

o: 170 10

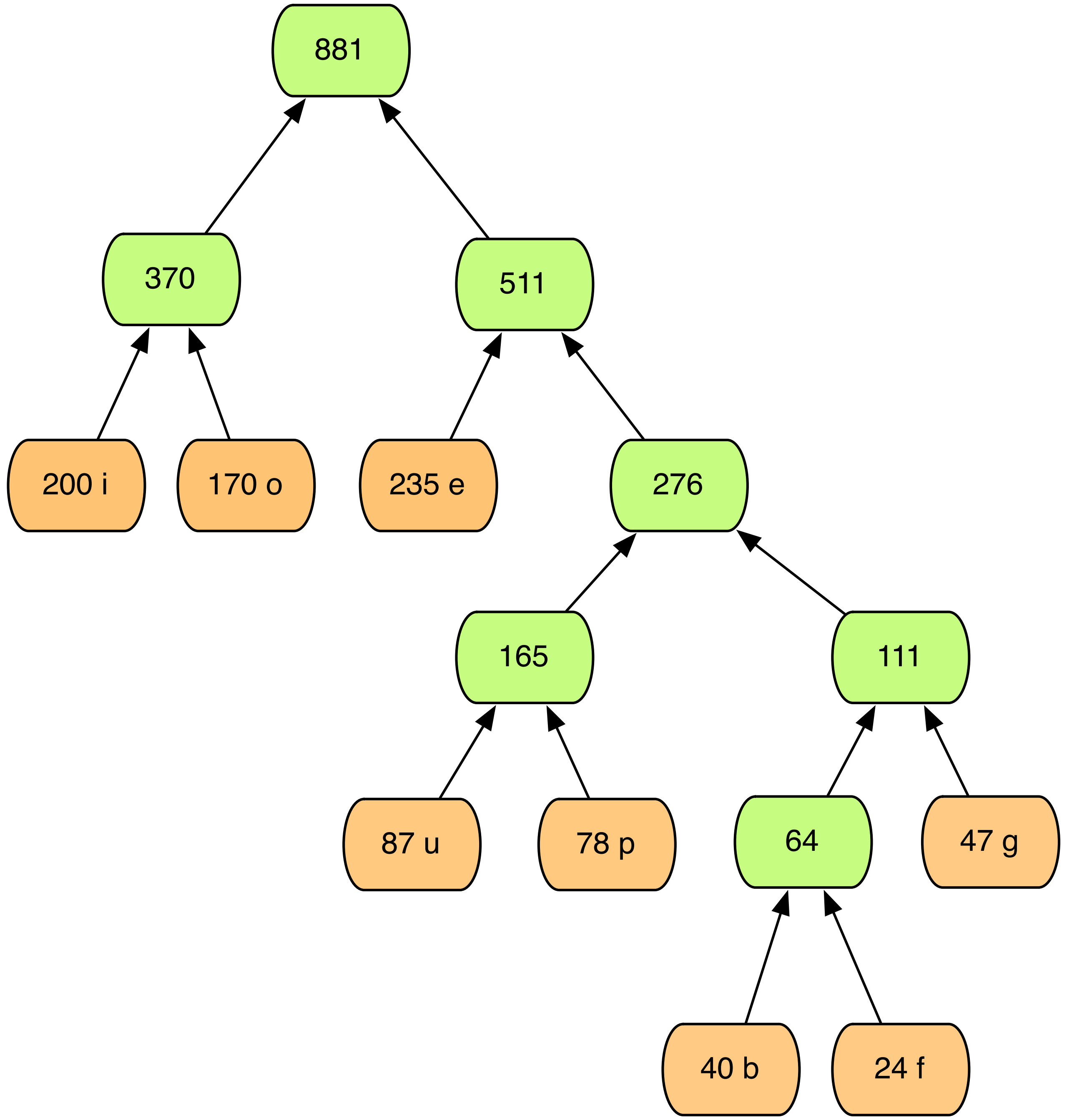
u: 87 0011

p: 78 0010

g: 47 0000

b: 40 00011

f: 24 00010



e:	235	01	470
i:	200	11	400
o:	170	10	340
u:	87	0011	348
p:	78	0010	312
g:	47	0000	188
b:	40	00011	200
f:	24	00010	120
			2378

2643

10%

objective

Prove that the Huffman algorithm produce the optimal prefix free code.

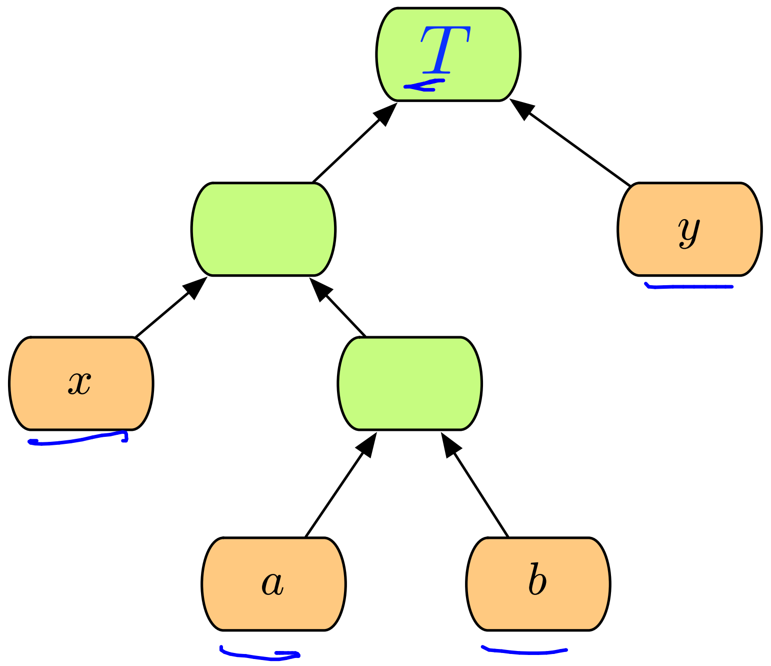
In search of an exchange argument.

Exchange argument

lemma: Let f_x and f_y be the 2 smallest frequencies in $\{f_c\}$.
There exists an optimal code such that x and y are siblings.

exchange argument

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



Proof: Let T be an optimal code for $\{f_c\}$.

If x and y are siblings in T, the claim holds.

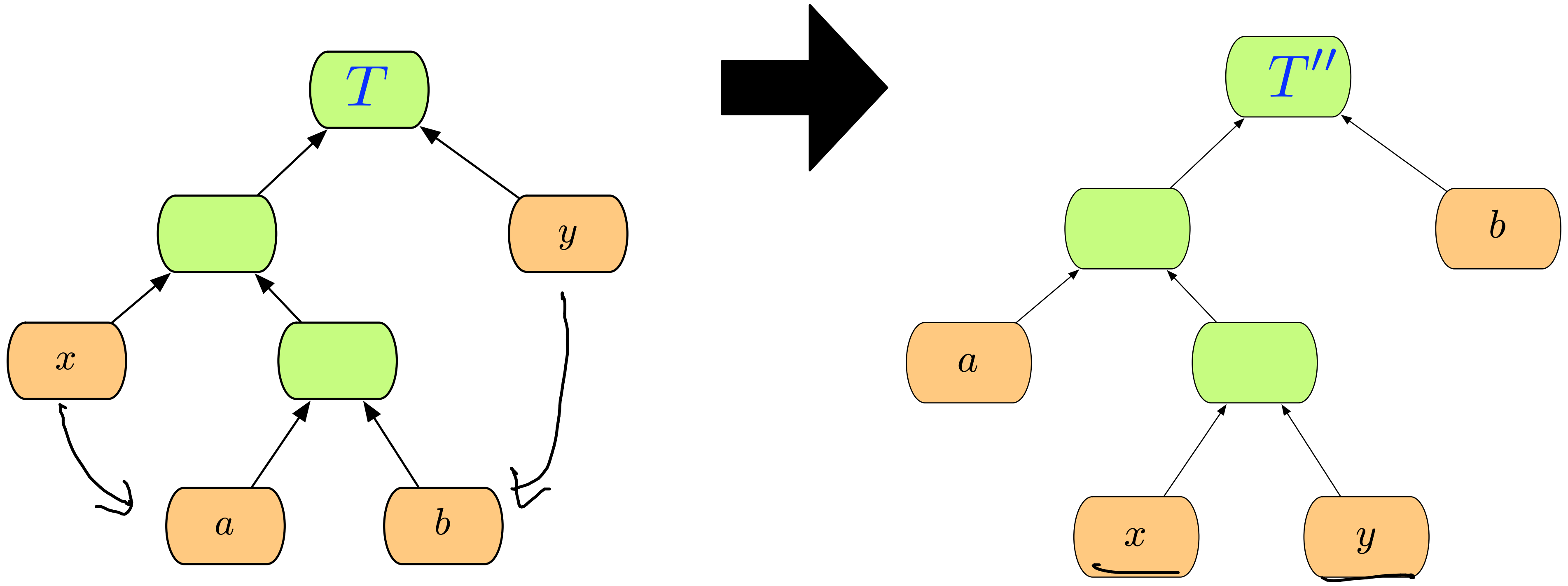
Otherwise, let a and b be the 2 symbols at greatest depth in T. (which are siblings)

① Why do a & b exist??

B/c T is optimal, & as we argued before, nodes only have 0 or 2 children in optimal codes.

exchange argument

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



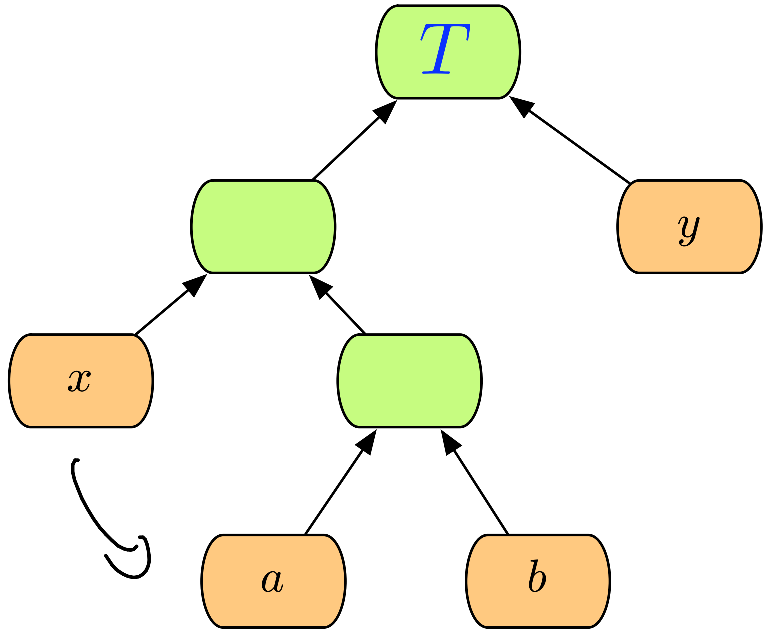
$f_x \leq f_a$ $f_y \leq f_b$

exchange argument

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

proof:

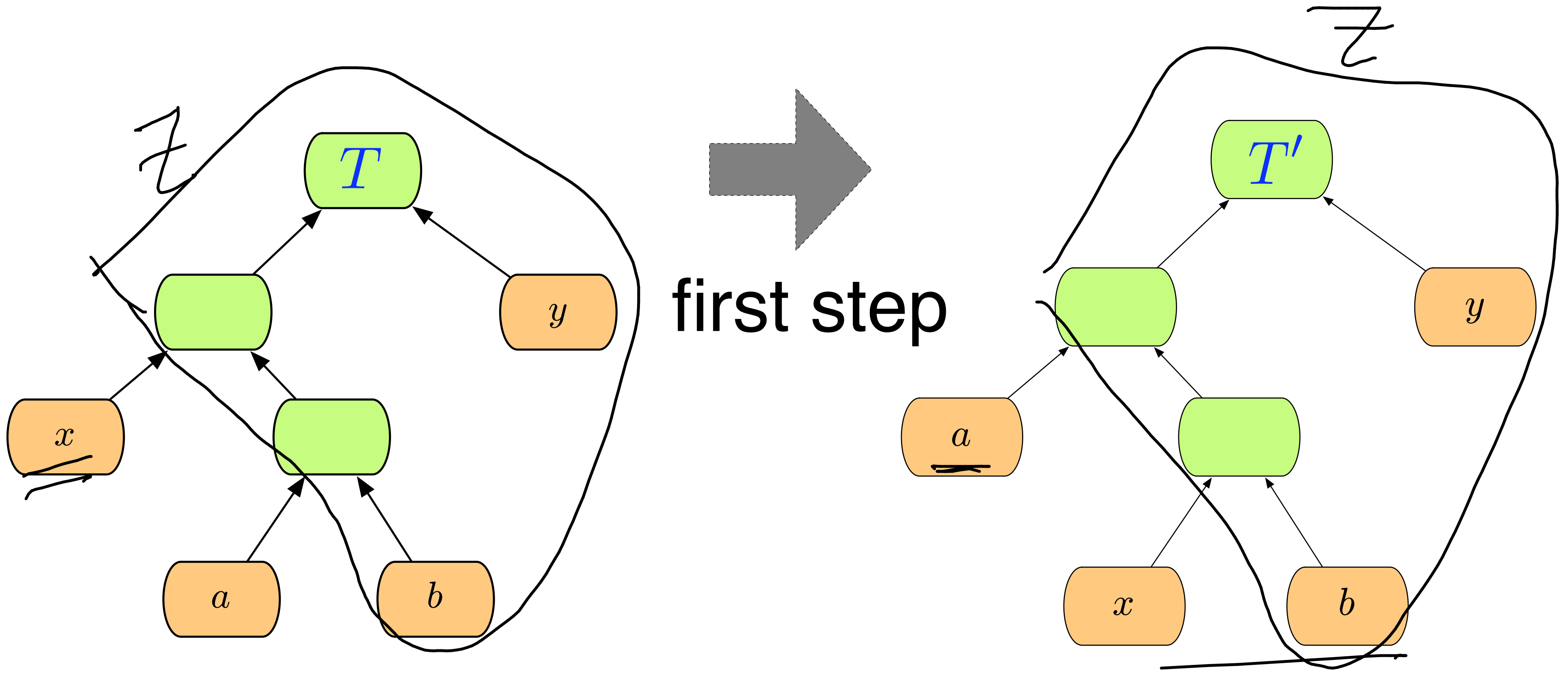
exchange argument



first step

Exchange a with x in the tree to construct a new tree T'.

exchange argument

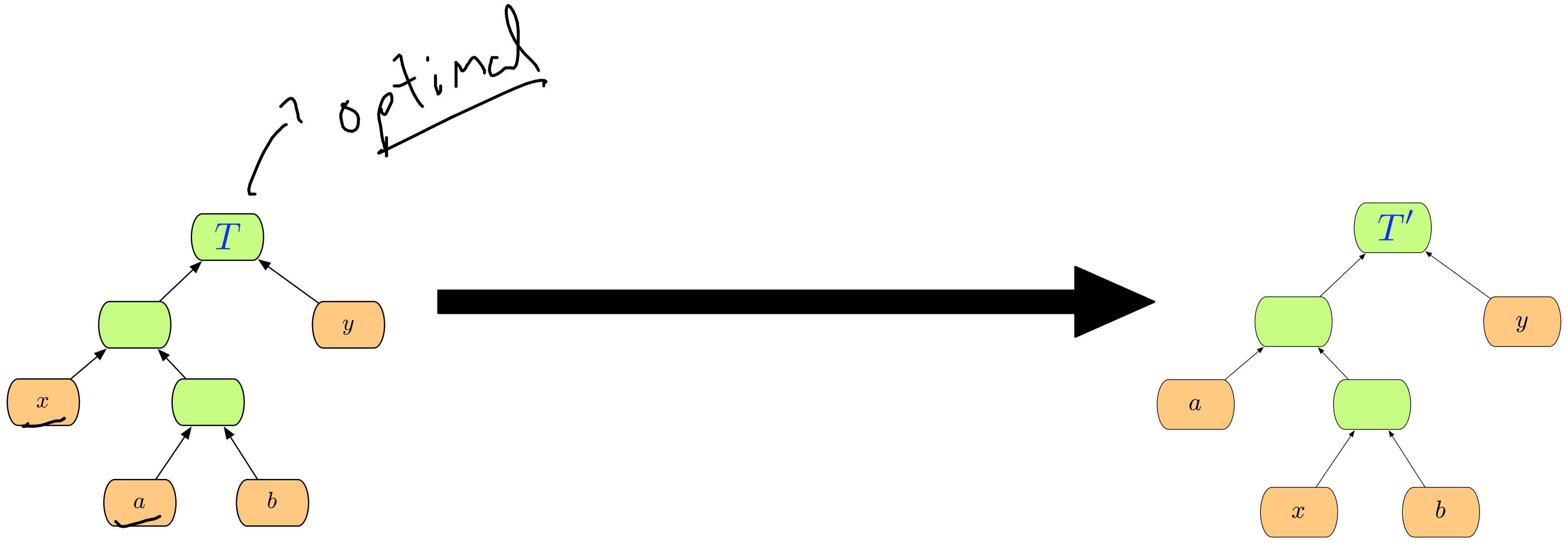


$f_a \leq f_b$ $\underline{f_x \leq f_a}$

$f_x \leq f_y$ $f_y \leq f_b$

$B(T) = Z + f_x \cdot l_x + f_a \cdot l_a$

$B(T') = Z + f_a \cdot l_x + f_x \cdot l_a$



$$B(T) = \cancel{Z} + \underline{f_x} \cdot l_x + f_a \cdot l_a$$

$$B(T') = \cancel{Z} + f_a \cdot l_x + \underline{f_x \cdot l_a}$$

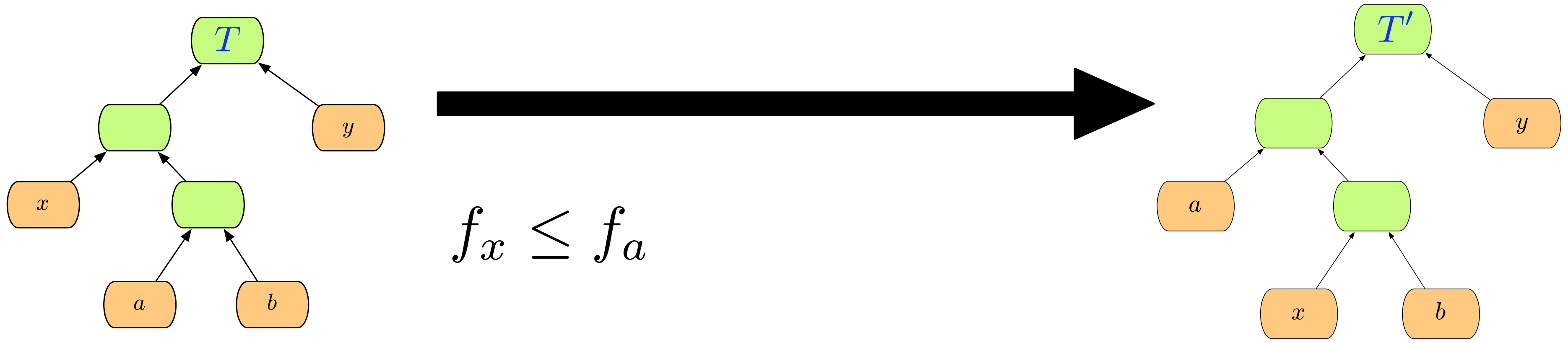
$$B(T) - B(T') = f_x(l_x - l_a) - f_a(l_x - l_a) \geq 0$$

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

$$f_x \leq f_a$$

$$l_x \leq l_a$$

But T is optimal, and so $B(T) = B(T')$

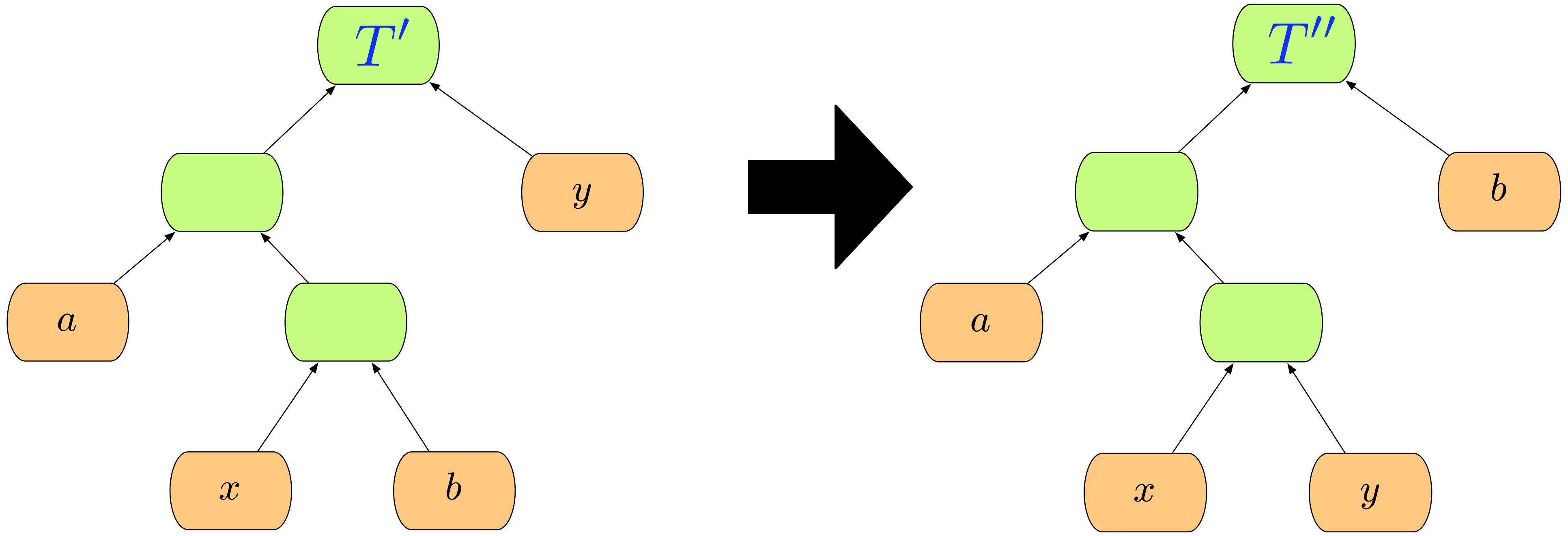


$$f_x \leq f_a$$

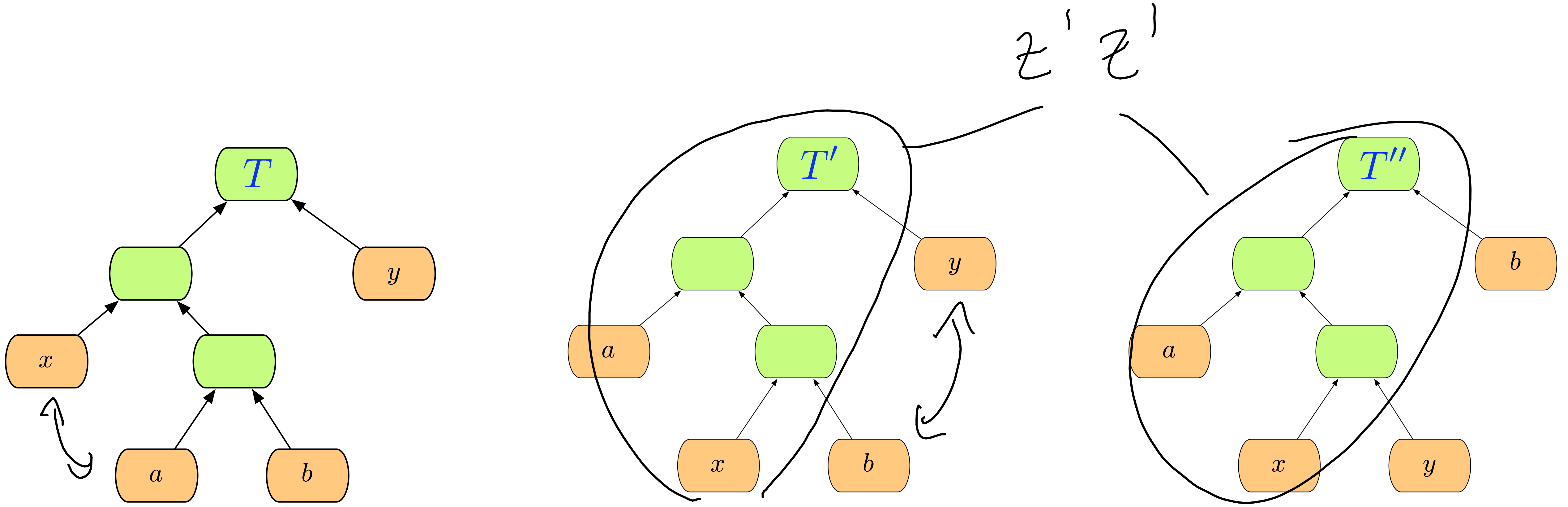
$$B(T) = \sum_c f_c \ell_c + f_x \ell_x + f_a \ell_a \qquad B(T') = \sum_c f_c \ell'_c + f_x \ell'_x + f_a \ell'_a$$

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

exchange argument



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

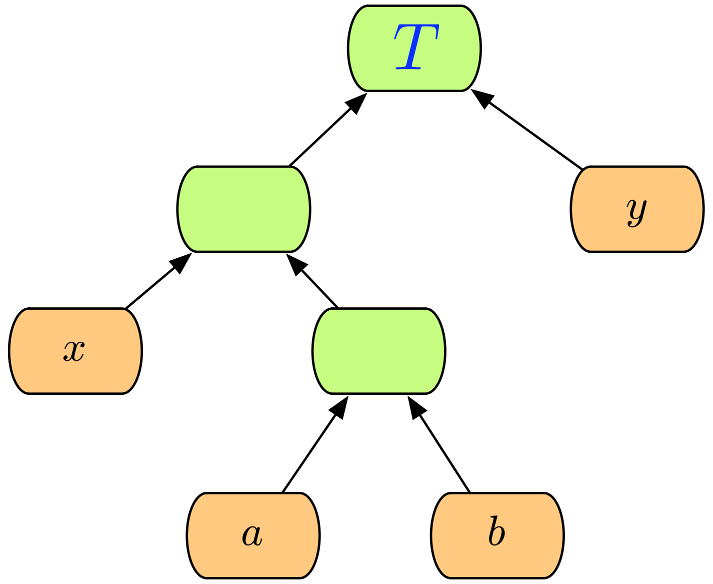


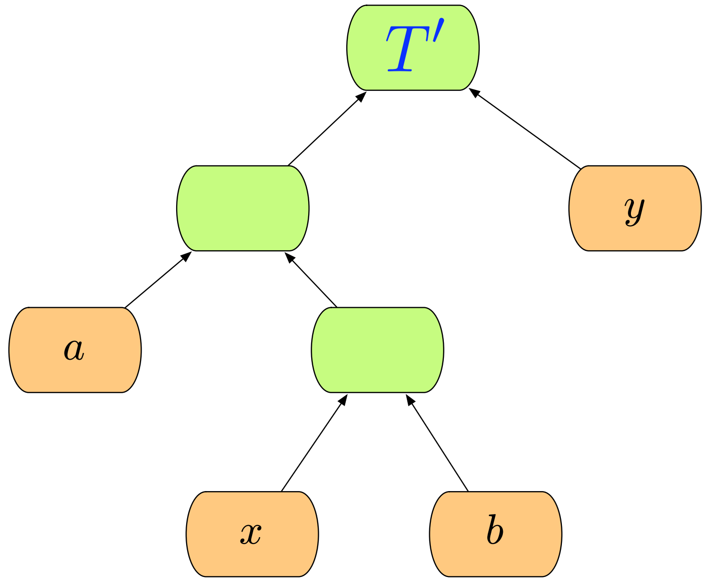
$B(T) - B(T') \geq 0$

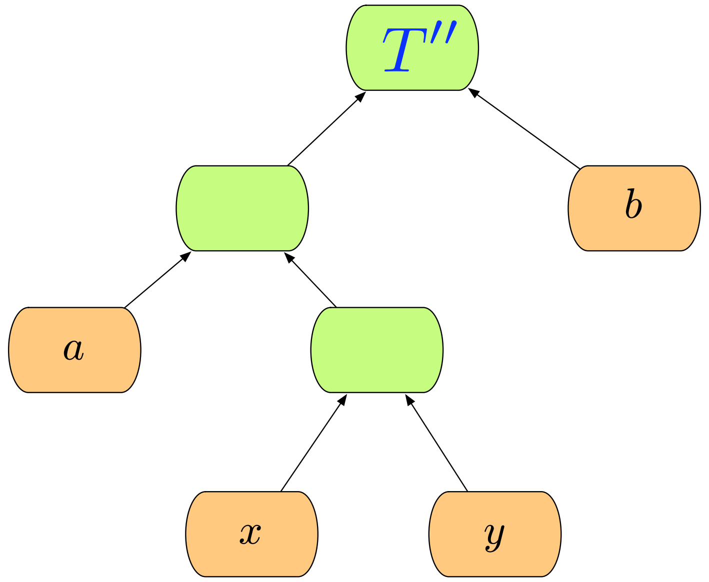
$B(T') - B(T'') \geq 0$

$B(T) - B(T'') \geq 0$, but again, b/c T is optimal,

they must be equal, ⇒ T'' is optimal.







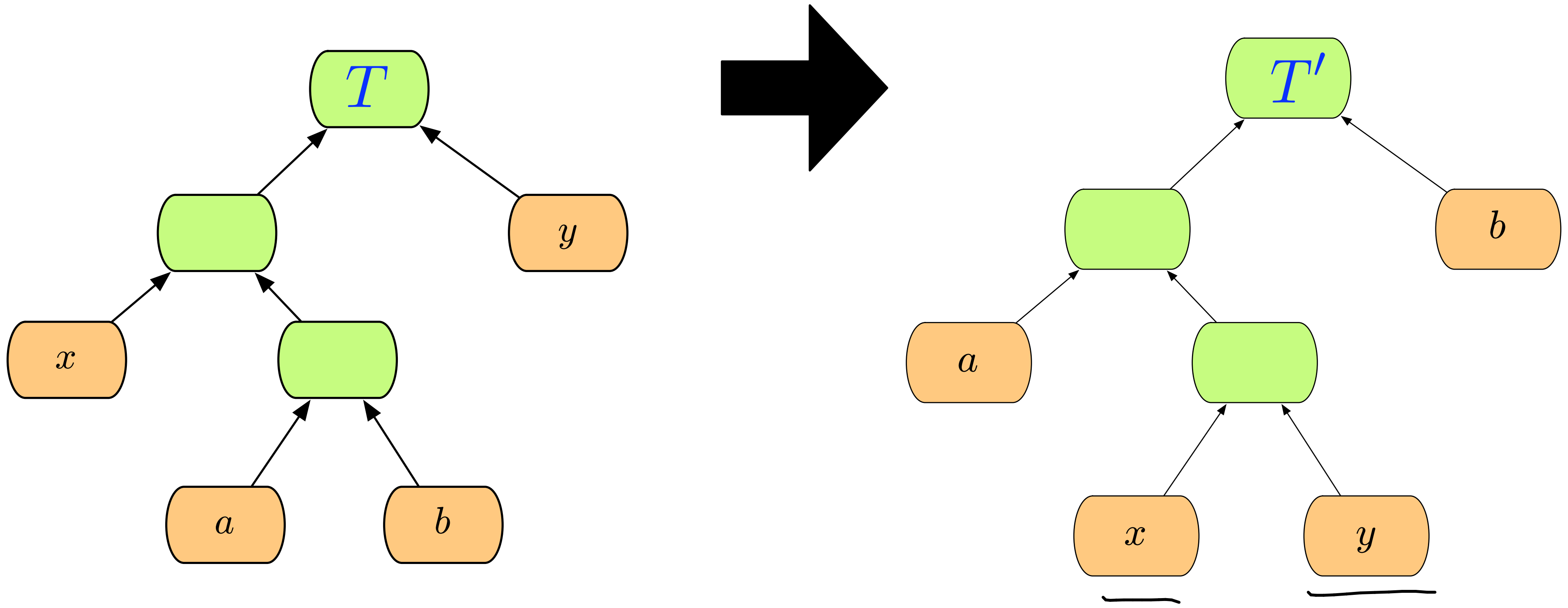
$B(T) - B(T') \geq 0$

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

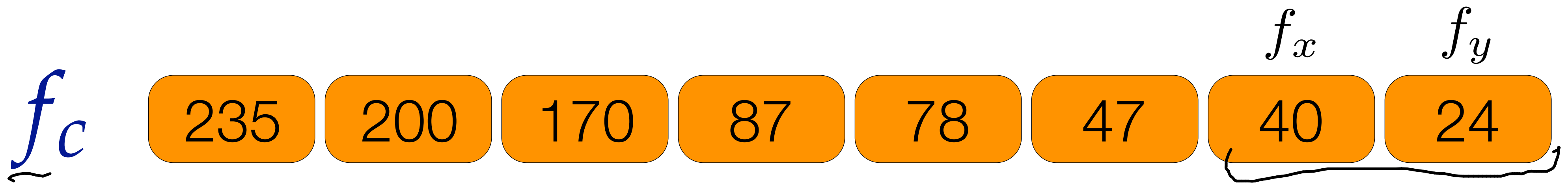
T'' is also optimal

exchange argument

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



optimal sub-structure



f_c

						f_x	f_y
235	200	170	87	78	47	40	24

problem of size n

optimal sub-structure

f_c

						f_x	f_y
235	200	170	87	78	47	40	24

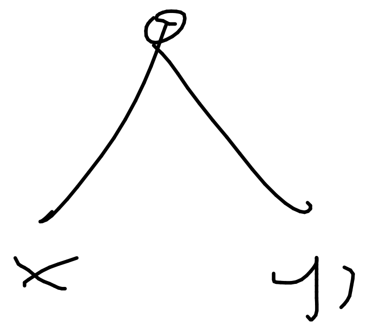
problem of size n

$f_{c'}$

235	200	170	87	78	47	64
						f_z

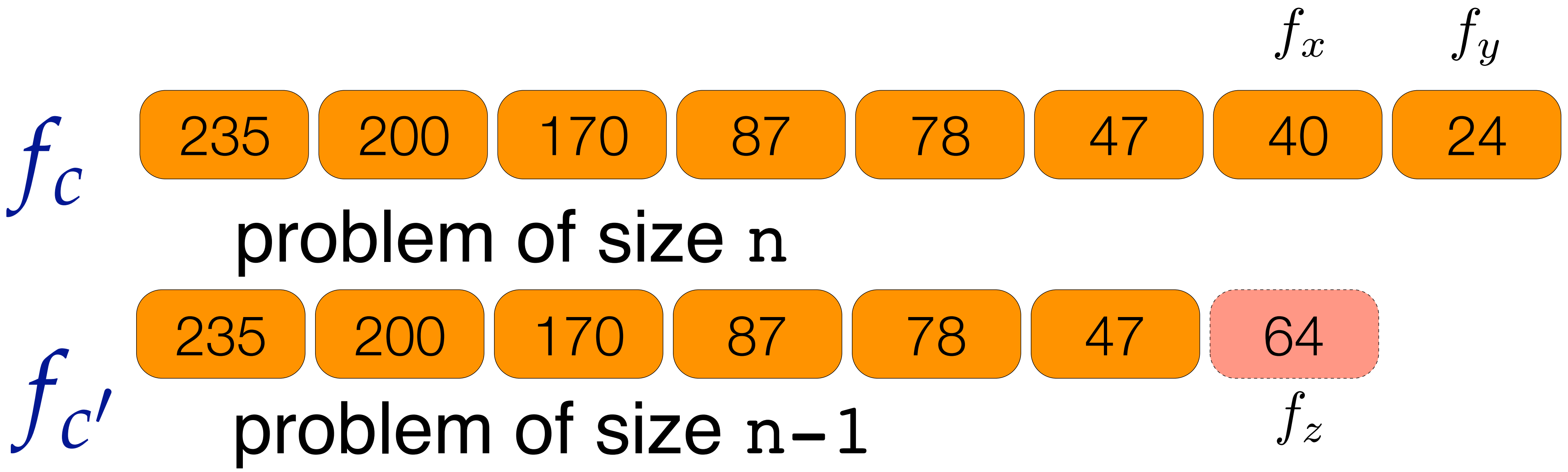
problem of size n−1

optimal solution to $f_{c'}$, and then replace f_z with



the resulting tree is optimal for f_c.

optimal sub-structure



f_x f_y

f_c

235	200	170	87	78	47	40	24

problem of size `n`

$f_{c'}$

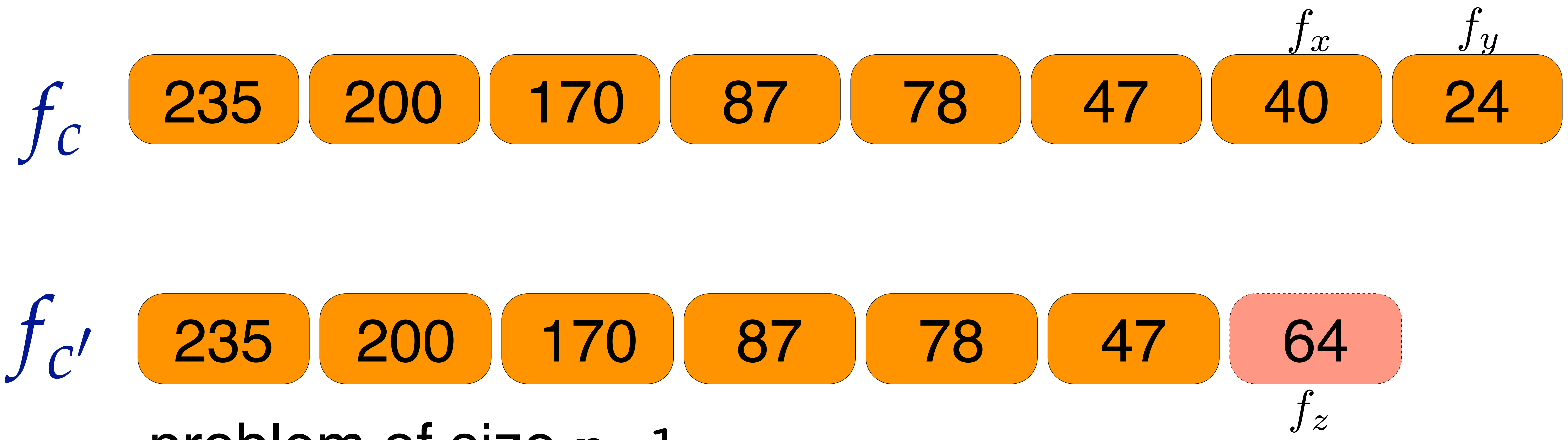
235	200	170	87	78	47	64

problem of size `n-1`

f_z

Lemma:

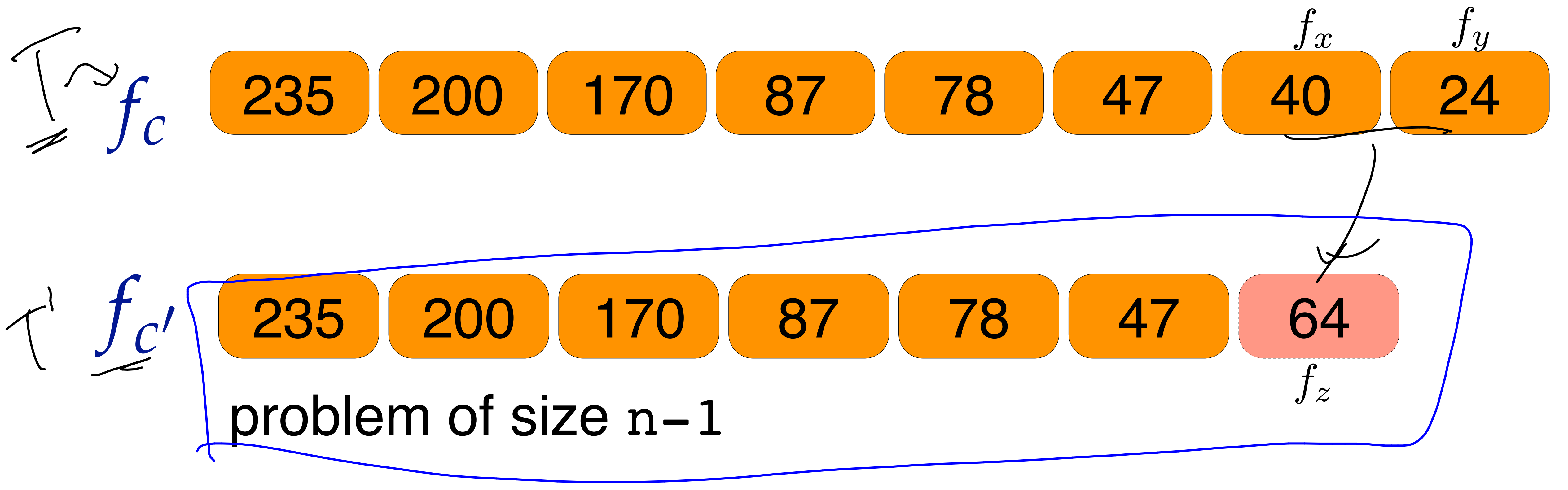
optimal sub-structure



							f_x	f_y
f_c	235	200	170	87	78	47	40	24
$f_{c'}$	235	200	170	87	78	47	64	
							f_z	

problem of size `n-1`

optimal sub-structure


T ~ f_c

f_x f_y

235 200 170 87 78 47 40 24

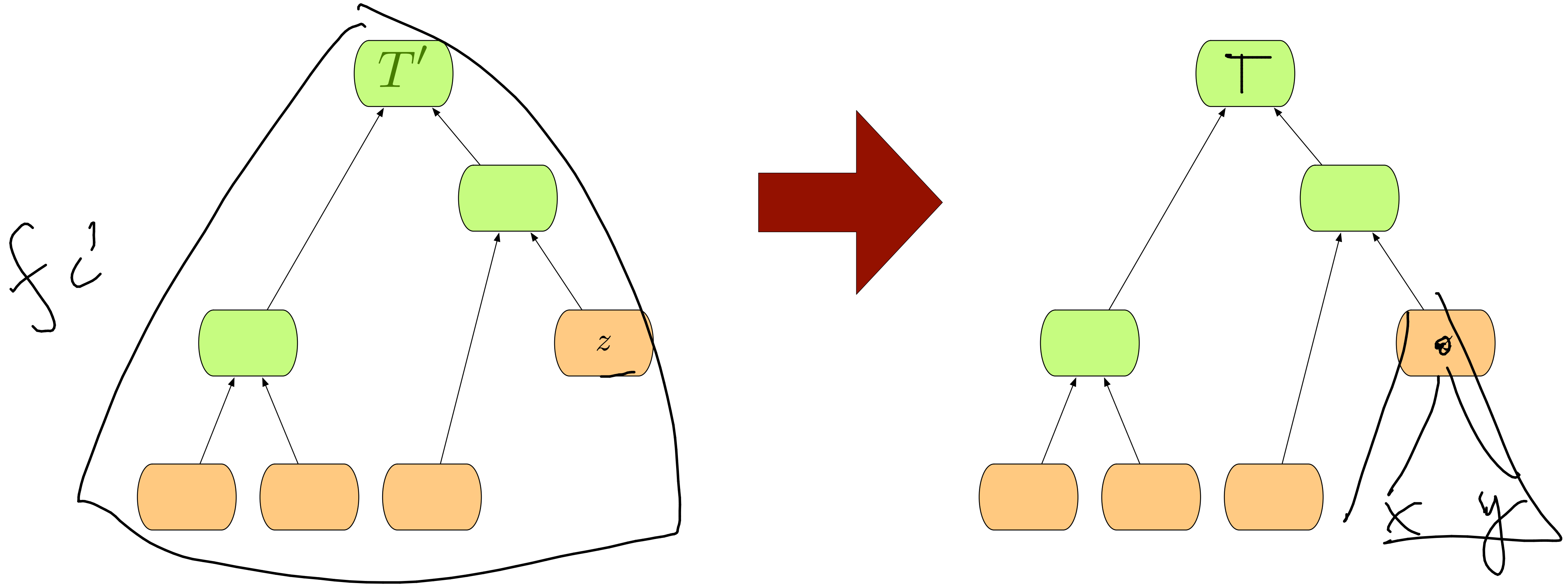
T' $f_{c'}$

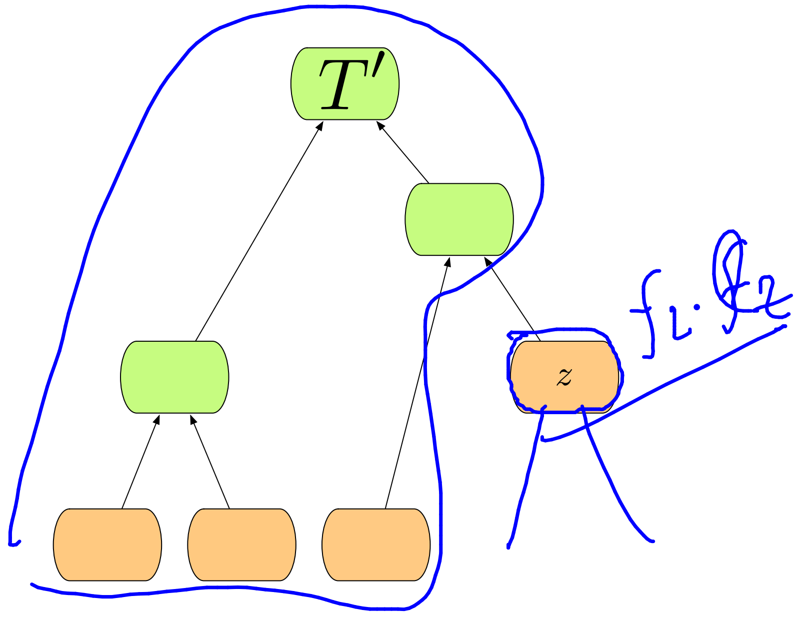
235 200 170 87 78 47 64

f_z

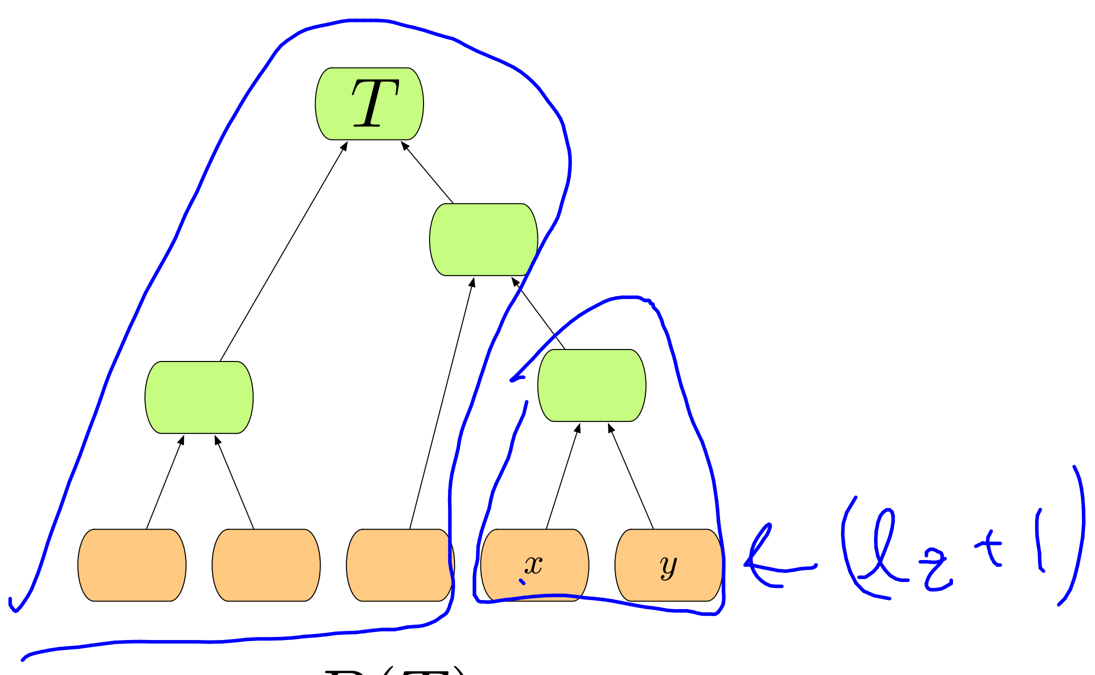
problem of size `n-1`

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.


fc'
T'
z
T
x
y



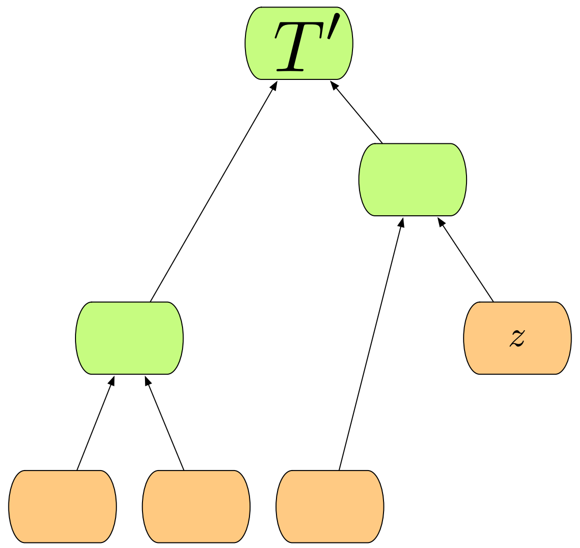
$B(T')$



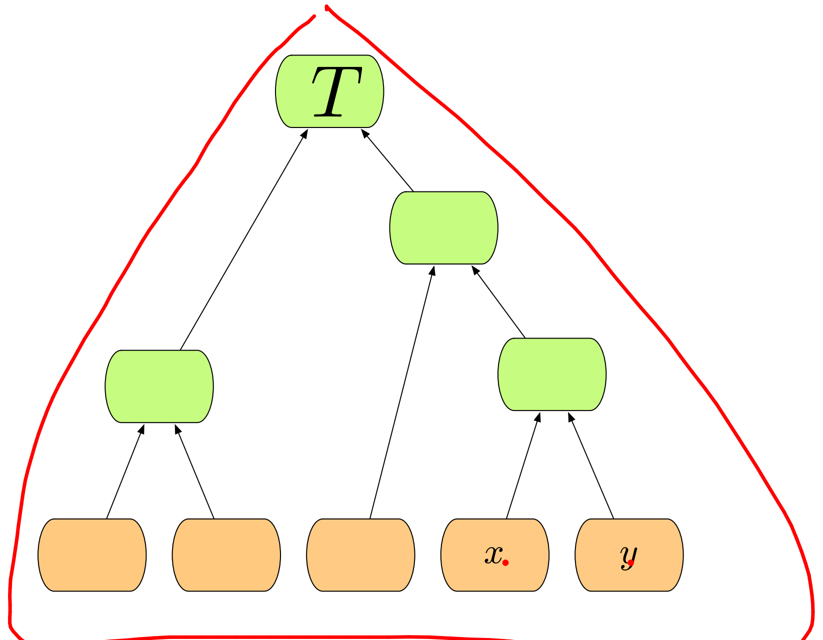
$B(T)$

$$B(T) = B(T') - f_z \cdot l_z + (l_z+1)(\overbrace{f_x + f_y}^{f_z})$$

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



$B(T')$



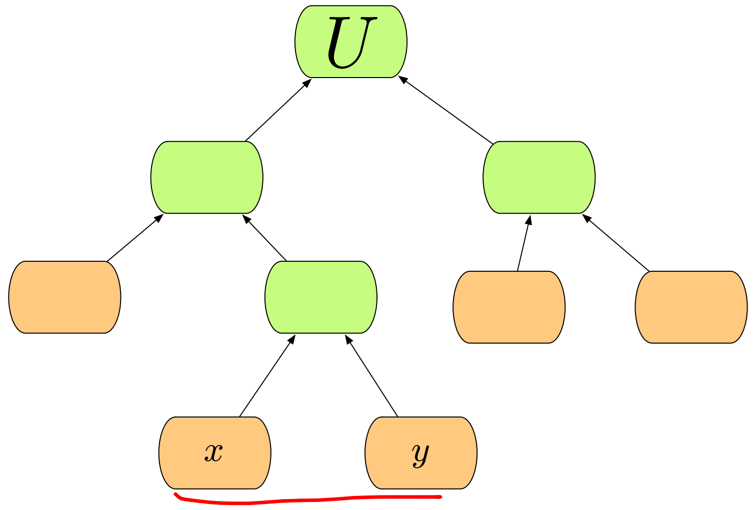
$B(T)$

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

Suppose T is not optimal.

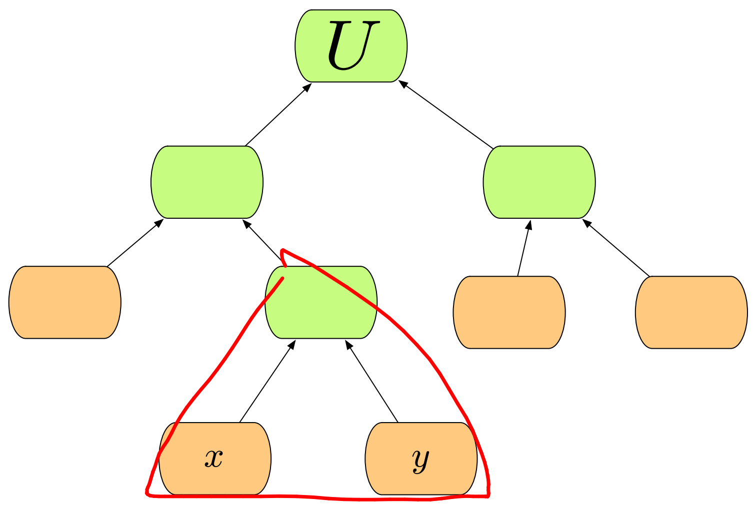
Suppose T is not optimal.

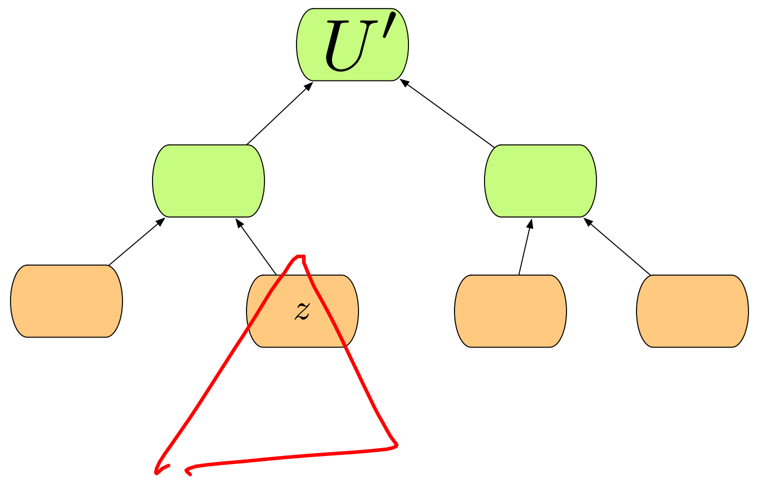
There is some other tree U such that



$$\underline{B(U)} < \underline{B(T)}$$

Suppose T is not optimal.





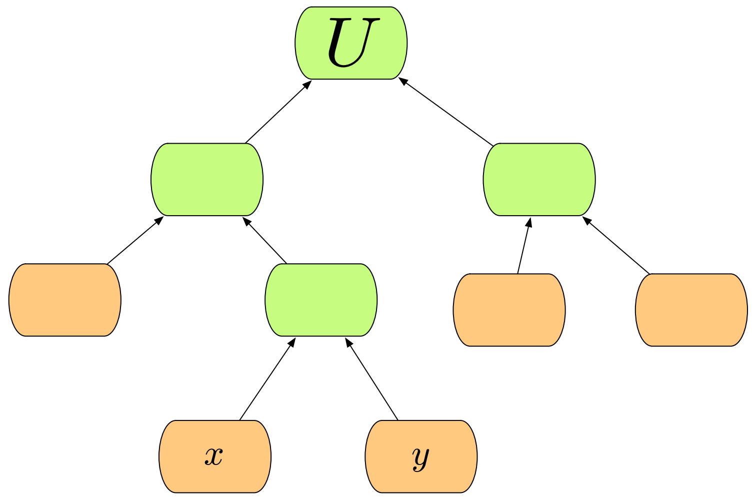
$$B(U) < B(T) = B(T') + f_x + f_y$$

$$B(U) - f_x - f_y < B(T')$$

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

This suggests that T' was not an optimal solution. This is a contradiction.

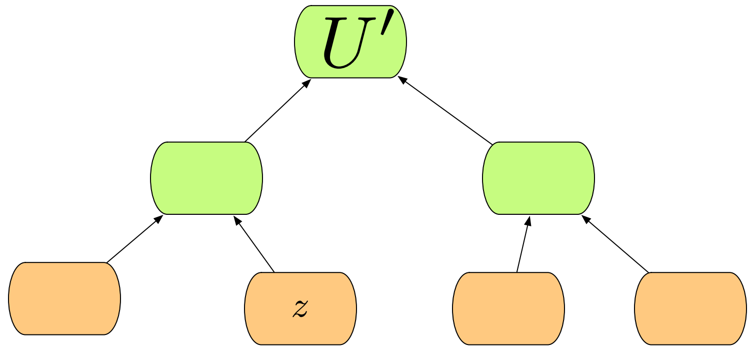
Suppose T is not



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

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

< B(t) - fx - fy



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

therefore