L15

4102

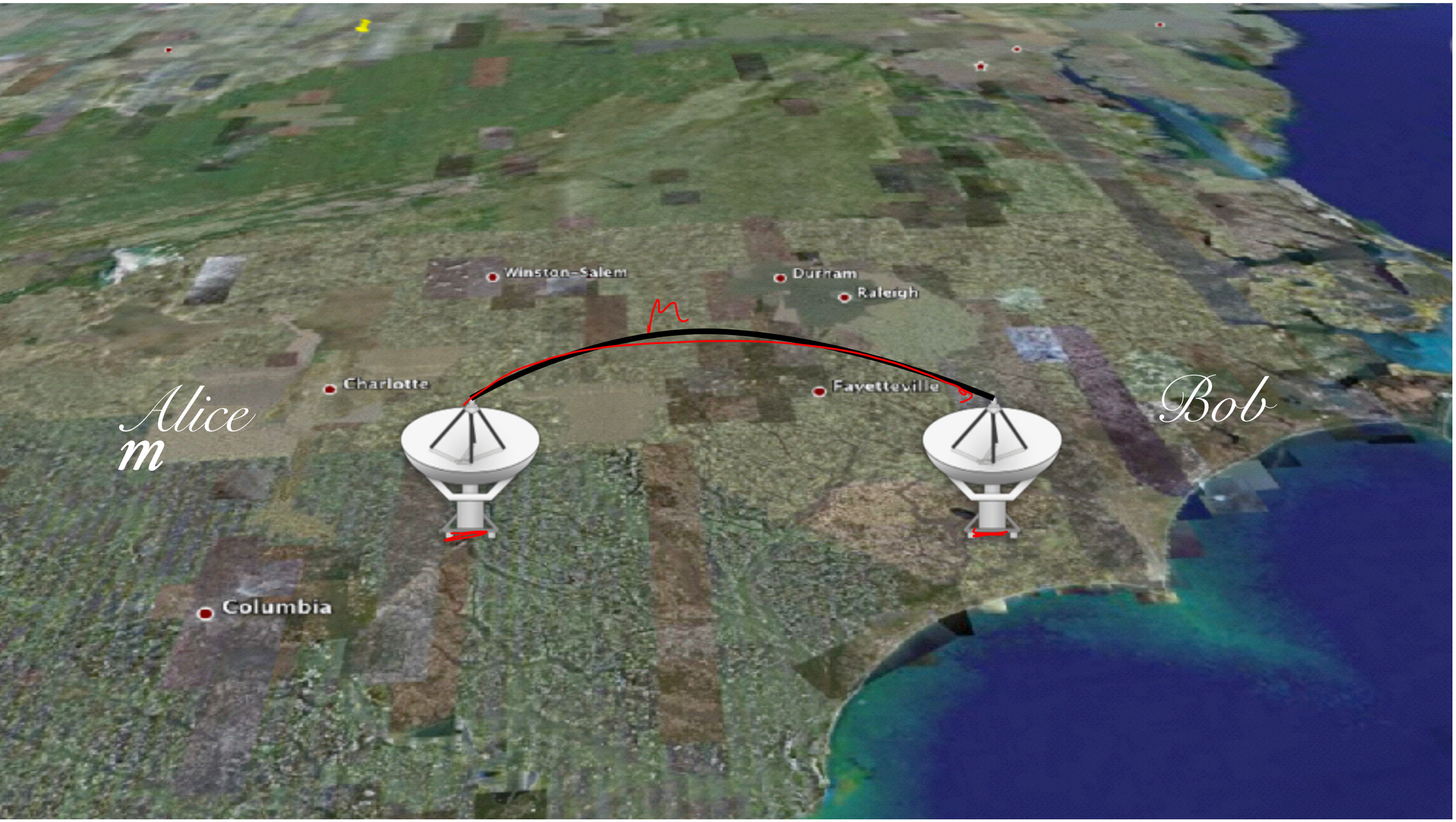
3.17.2016

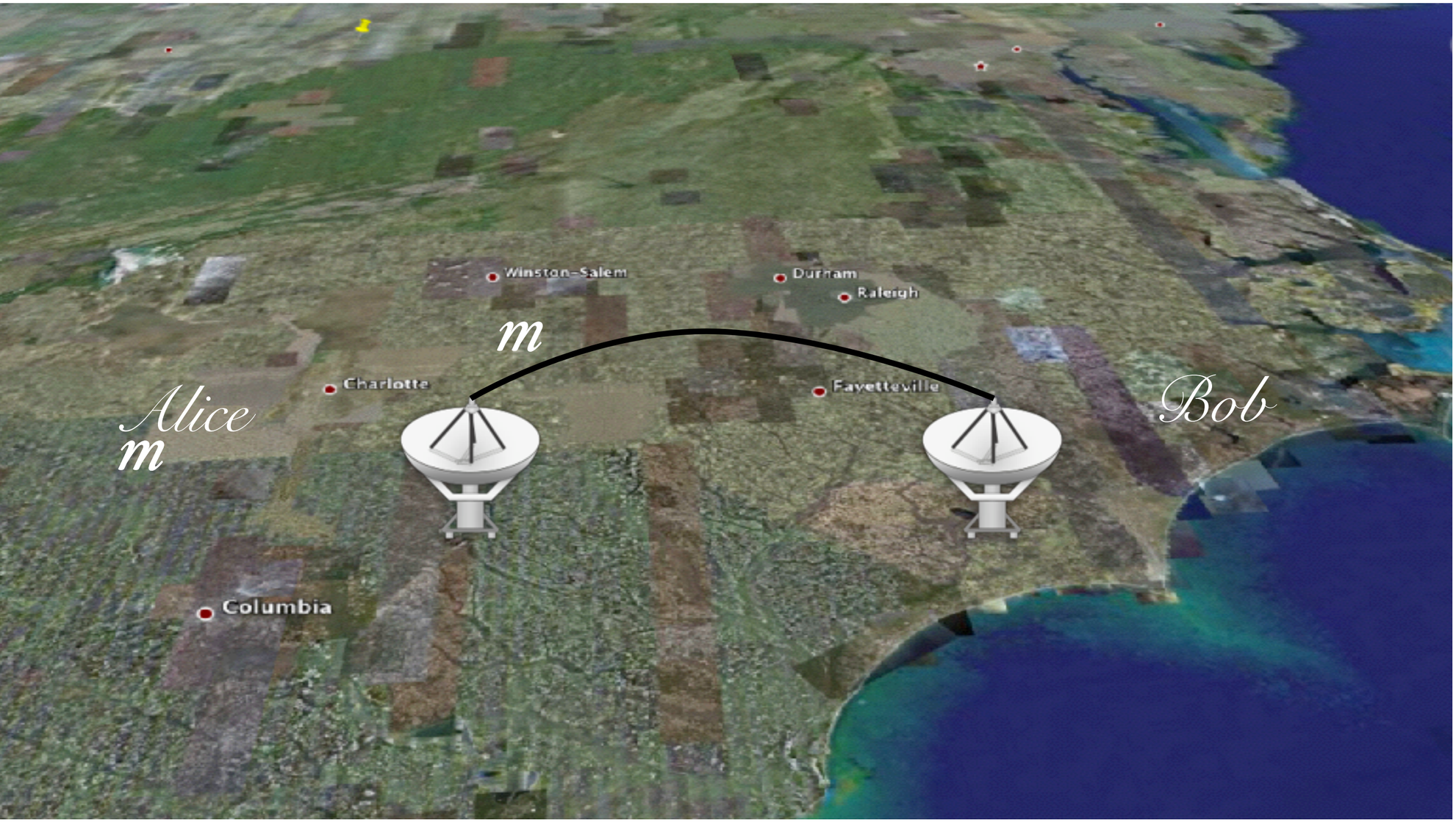
abhi shelat

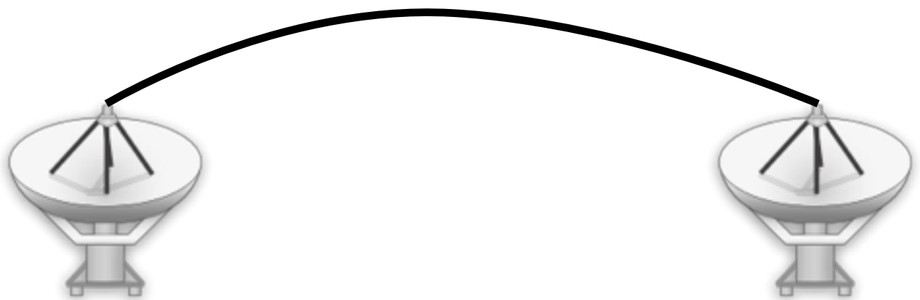
Huffman

image: wikimedia

SAM MORSE

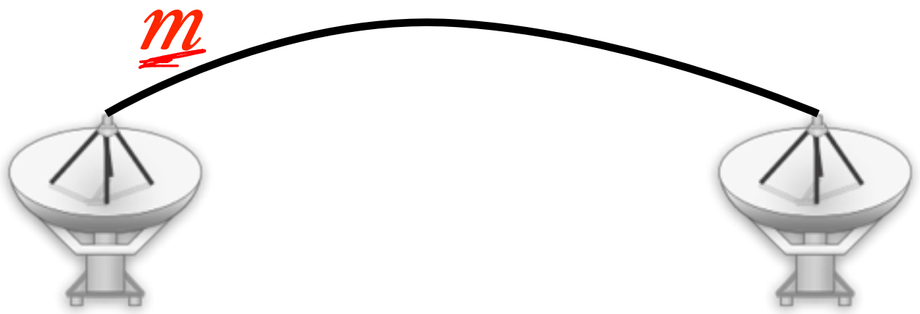

Winston-Salem
Durham
Raleigh
Charlotte
Fayetteville
Columbia
Alice
m
Bob


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



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



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

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

881

frequency

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 . 3 = 2643

def: cost of an encoding

encoding

frequency for each $c \in C$

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

tree

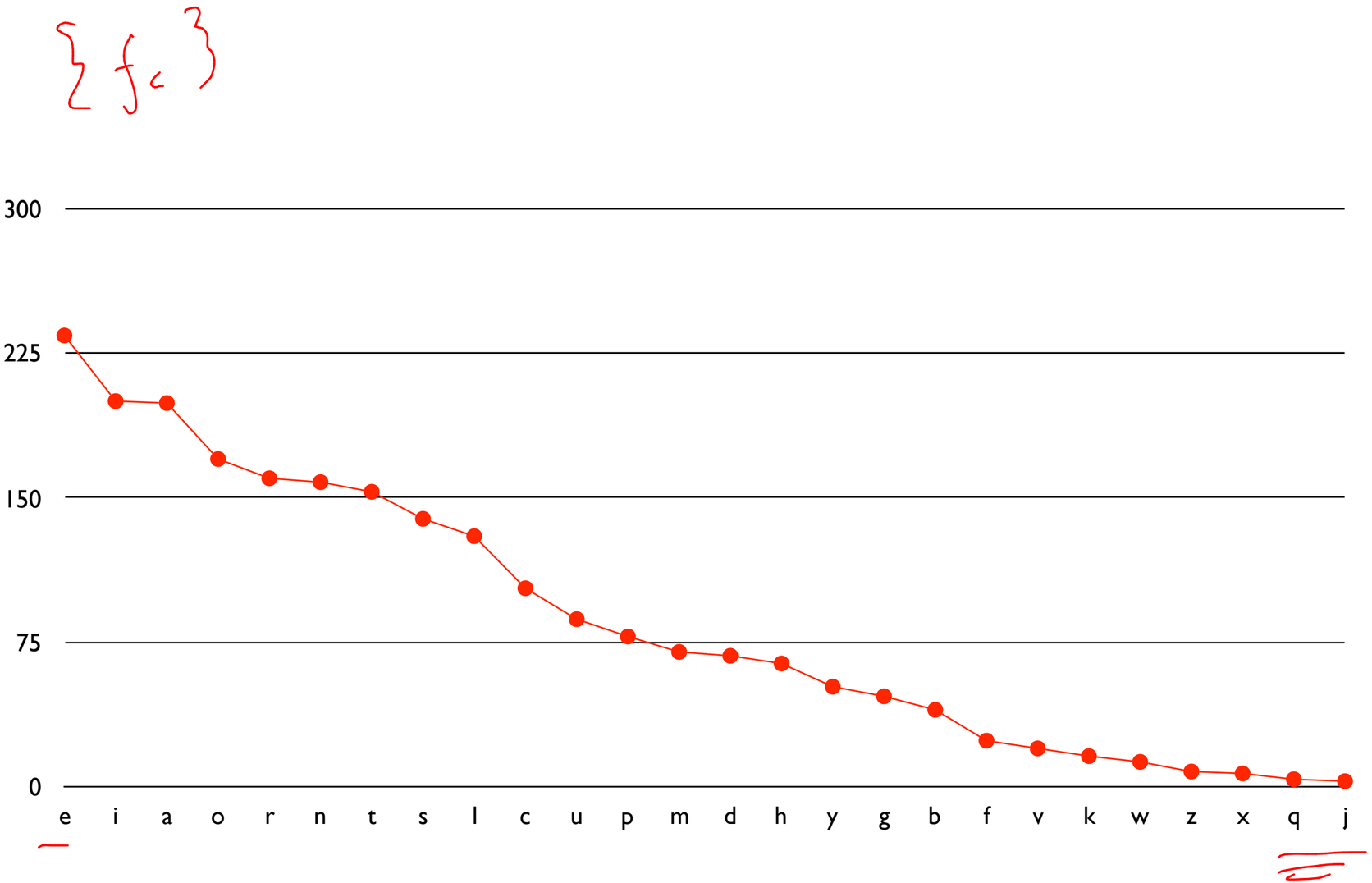
$c \in C$	f_c	T	ℓ_c
e:	235	000	3
i:	200	001	3
o:	170	010	3
u:	87	011	3
p:	78	100	3
g:	47	101	3
b:	40	110	3
f:	24	111	3
	881	·	3 = 881·3

$235 \cdot 3 + 200 \cdot 3 + 170 \cdot 3 + \cdots +$

$24 \cdot 3 = 881 \cdot 3 = 2643$

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

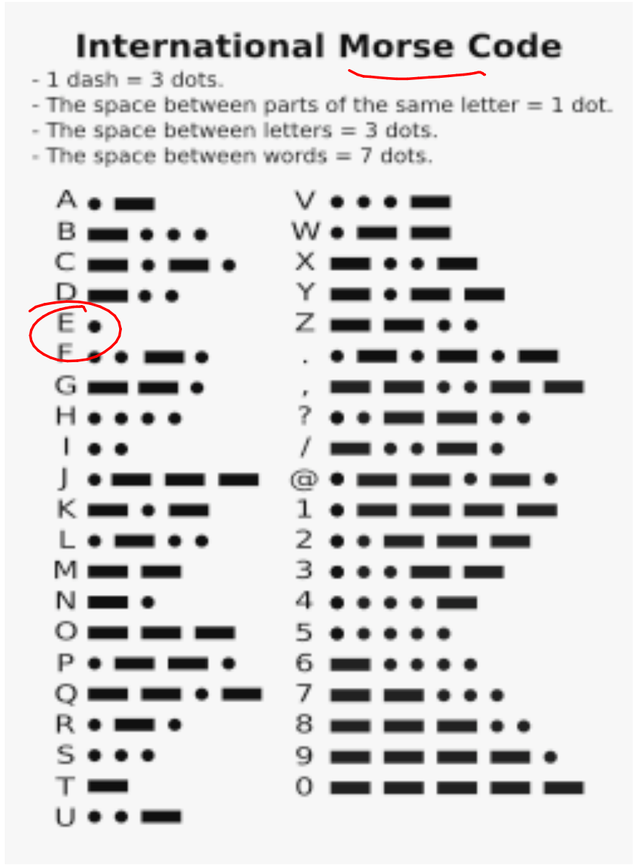
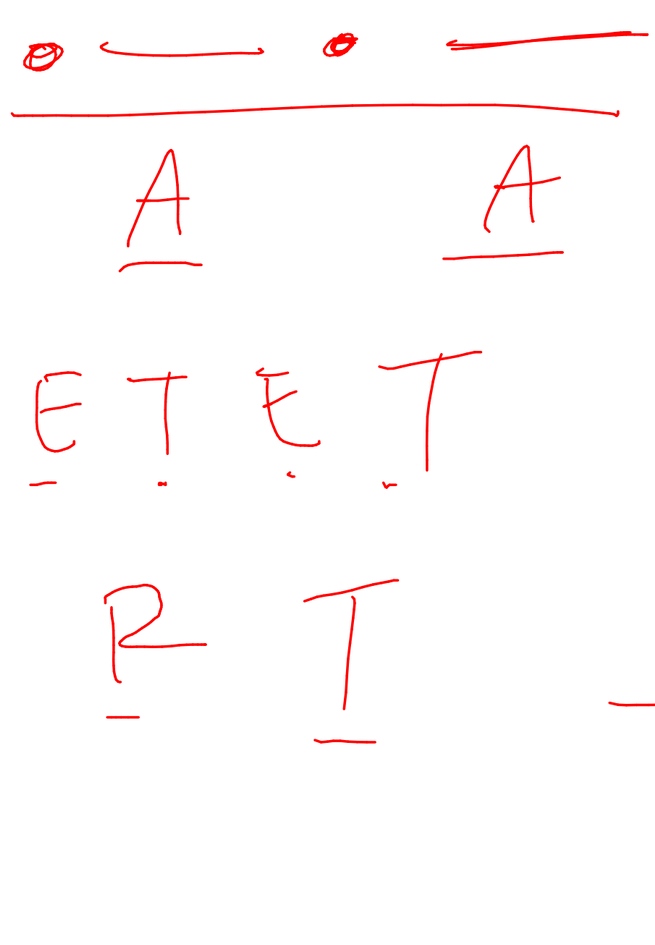


image http://en.wikipedia.org/wiki/Morse_code

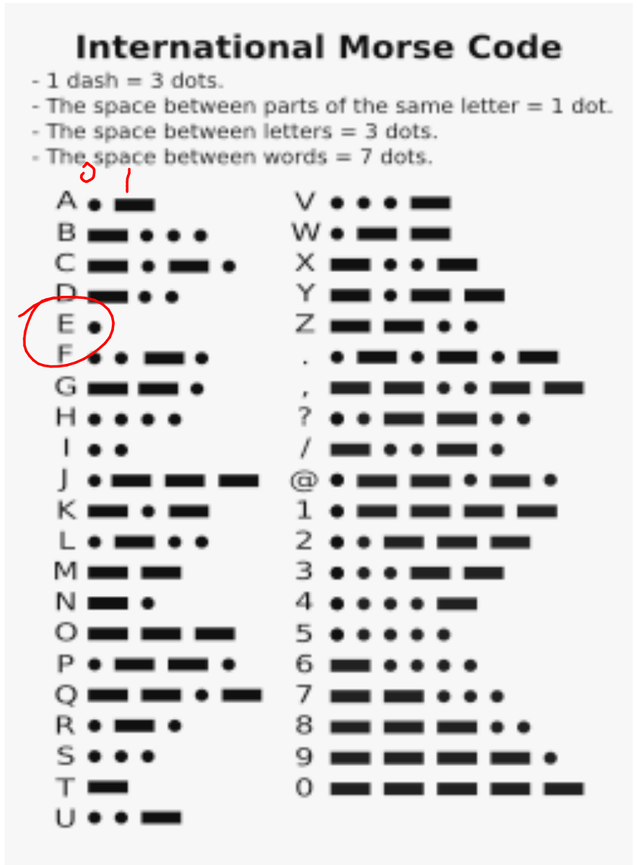


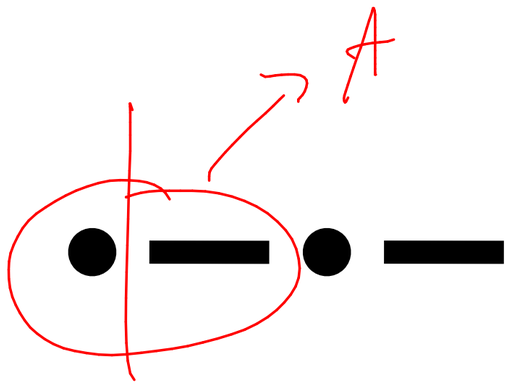
code

decode to several messages

— this is a problem

Morse code





encoding e - 0 Is A PREFIX of the encoding for A.

def: prefix-free code

Code for C such that for any two $x, y \in C$

if $x \neq y$ then code(x) is NOT A prefix of code(y).

def: prefix-free code

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

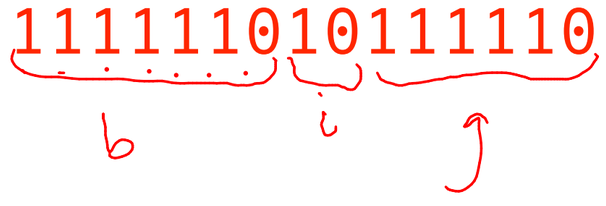
def: prefix code

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

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

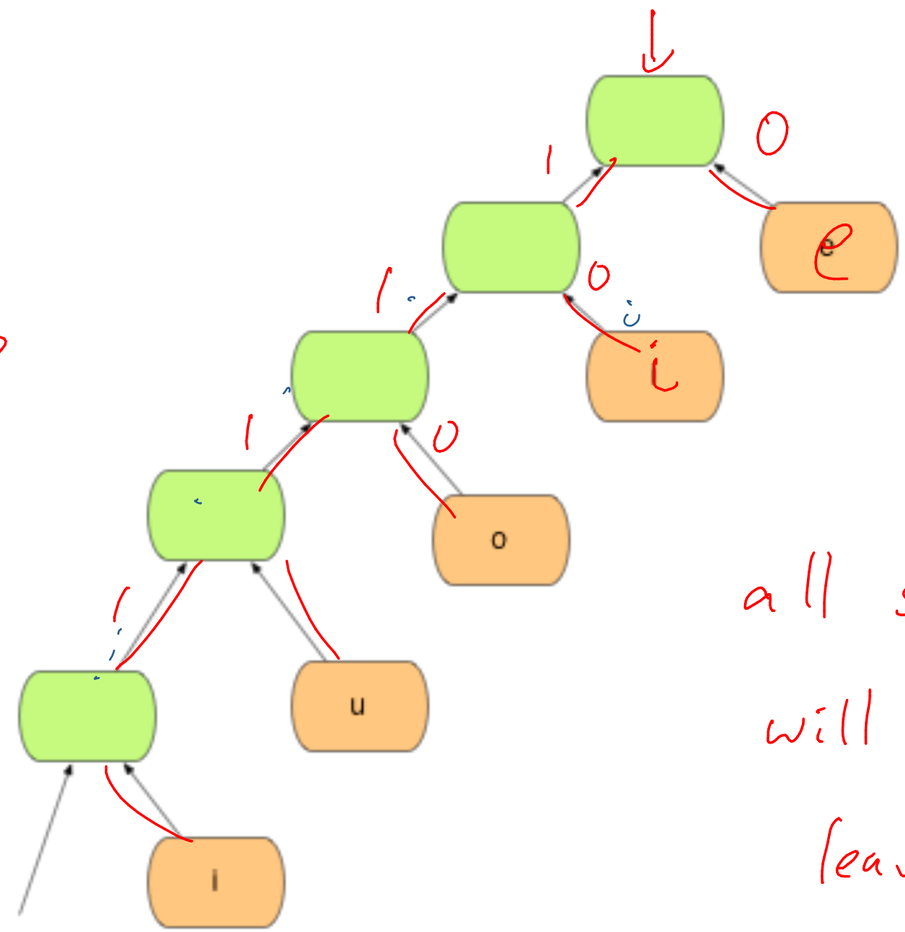
decoding a prefix code

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



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

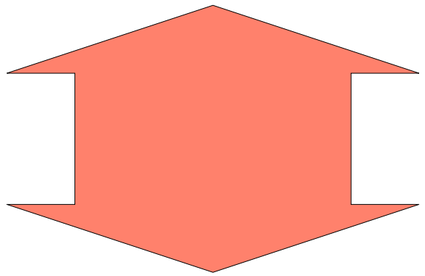


all symbols in C will be unique leaves in the tree.

1111110 10111110

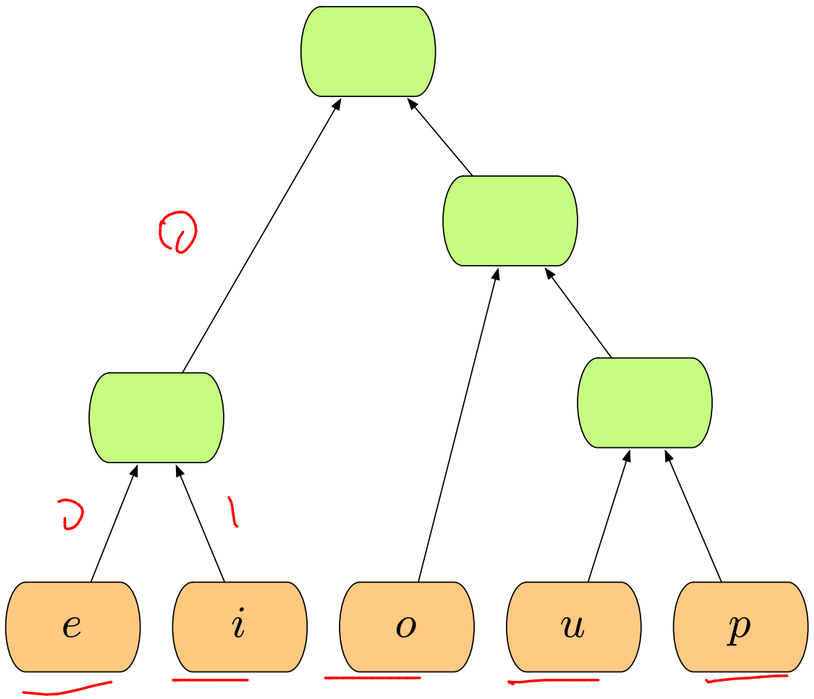
prefix code

free



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

input

given the frequencies $\{f_c\}$ for an alphabet C,

find the optimal prefix-free code T → output

that minimizes

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

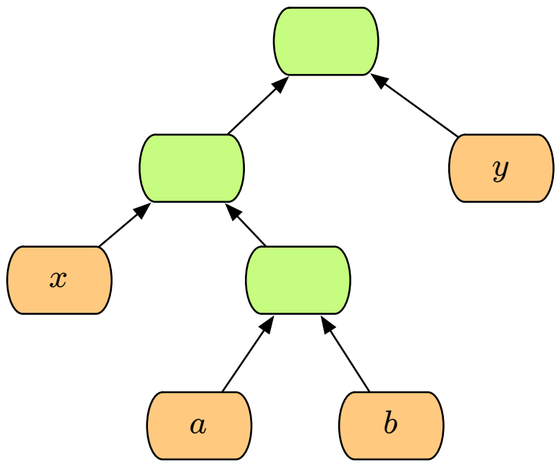
Goal

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

produce a prefix code T with smallest cost

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

Property

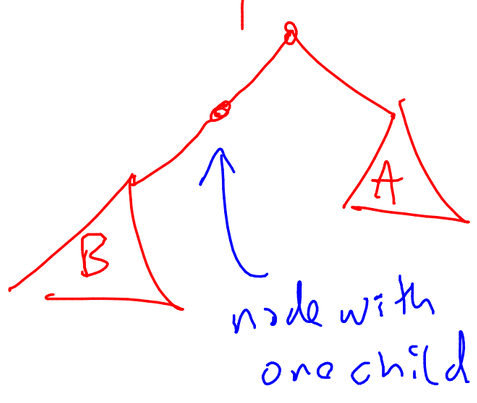


assume all frequencies > 0.

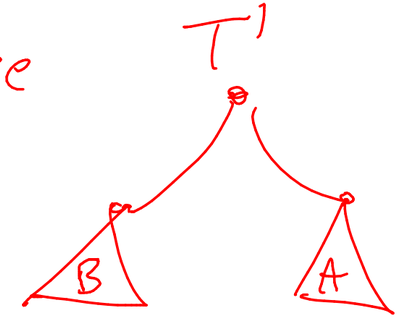
Lemma:Optimal tree is full.

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

Proof: Spse optimal T has a node with $\neq$ child



Consider tree



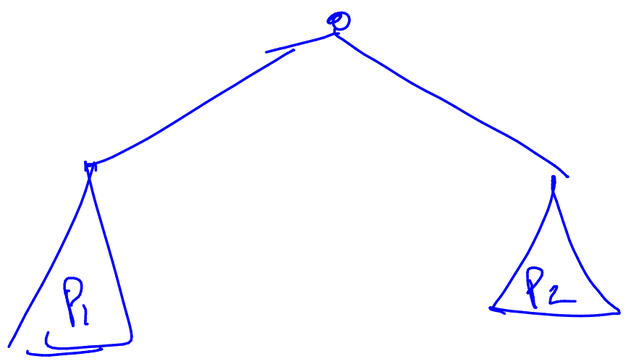
claim is that

$B(T) > B(T')$

because all symbols in subtree B have encodings that are 1 bit longer than in T'.

Divide & conquer?

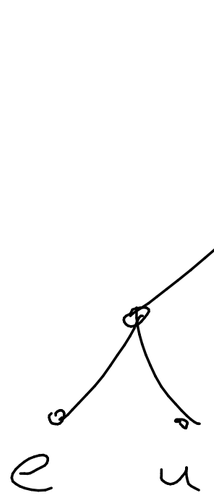
Partition the frequencies into 2 sets
P_1 P_2. Build trees for these & combine.



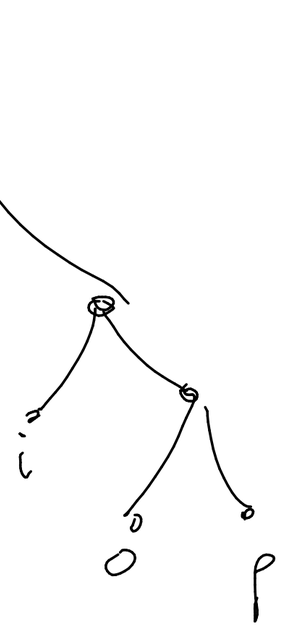
counter-example

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

e, u = 50

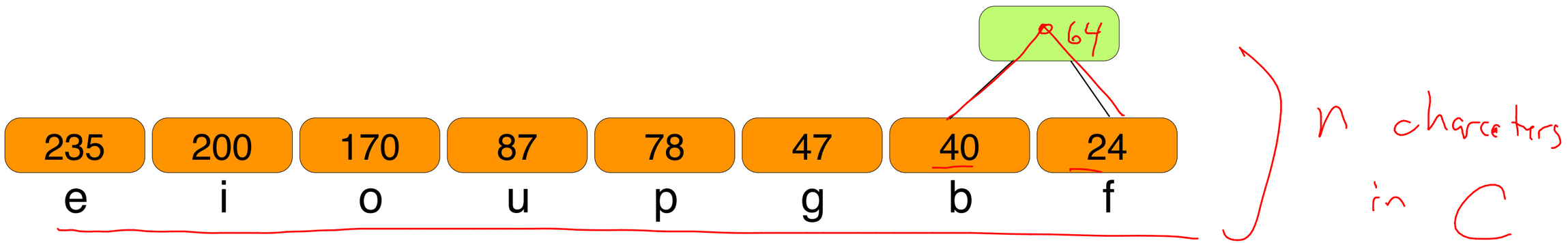


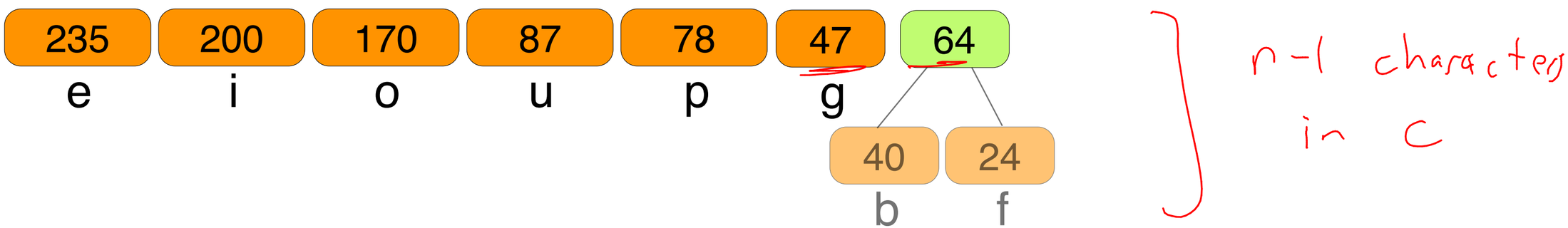
i, o, p - 50

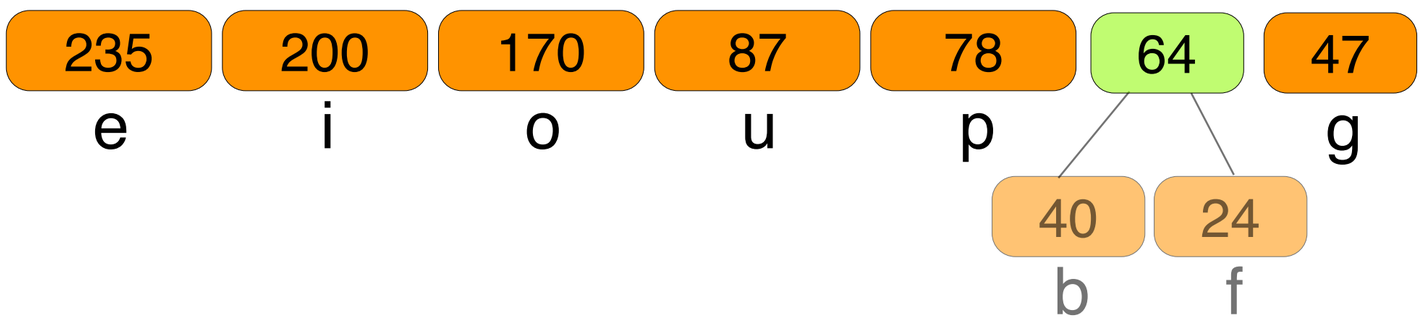


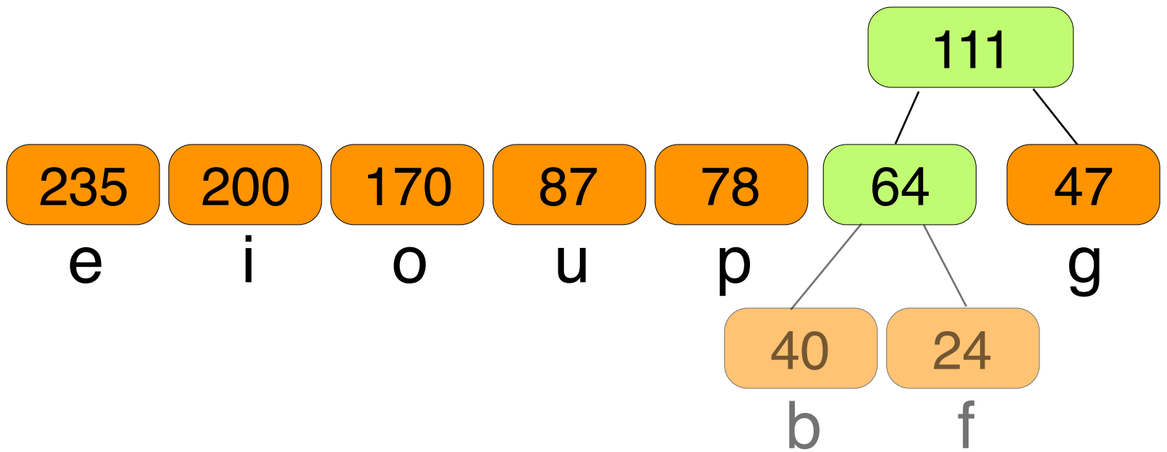
e - 00
u - 01
i - 10
o - 110
p - 111

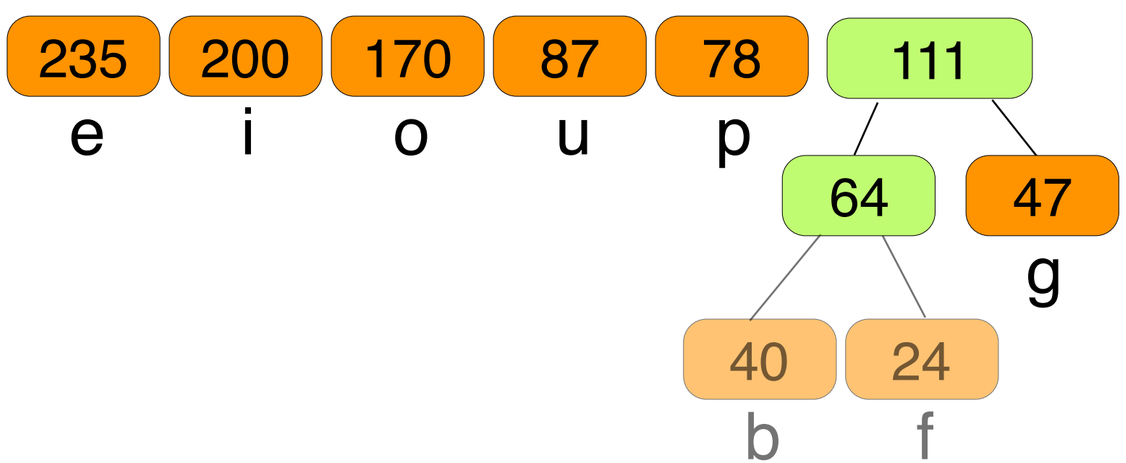
swapping the codes for o and u results in a better code b/c $f_o > f_u$

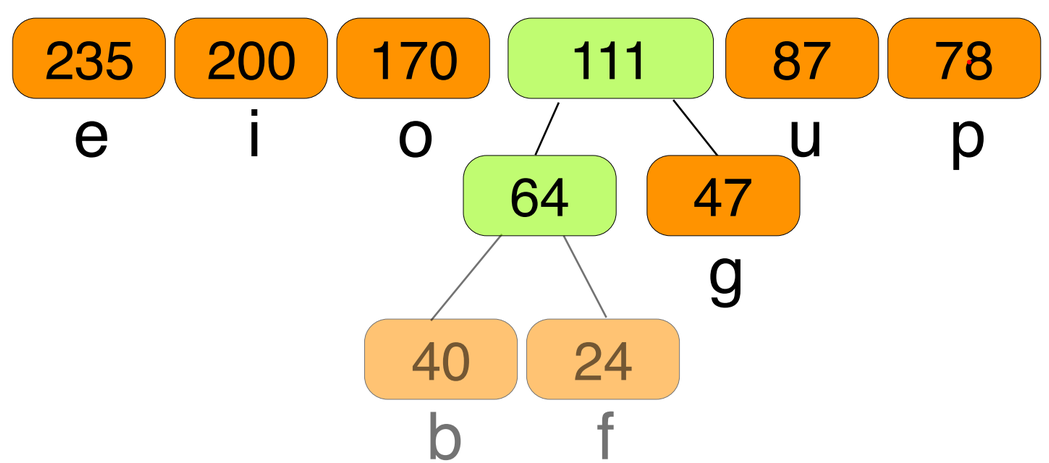

64
235
200
170
87
78
47
40
24
e
i
o
u
p
g
b
f
n characters
in C

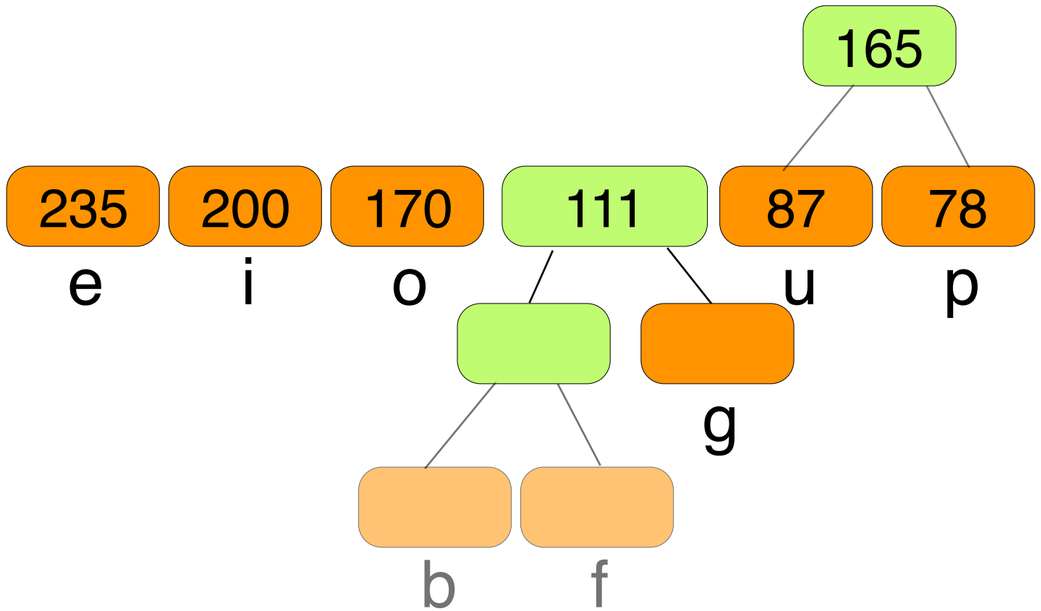

235
200
170
87
78
47
64
e
i
o
u
p
g
40
24
b
f
n-1 characters in C

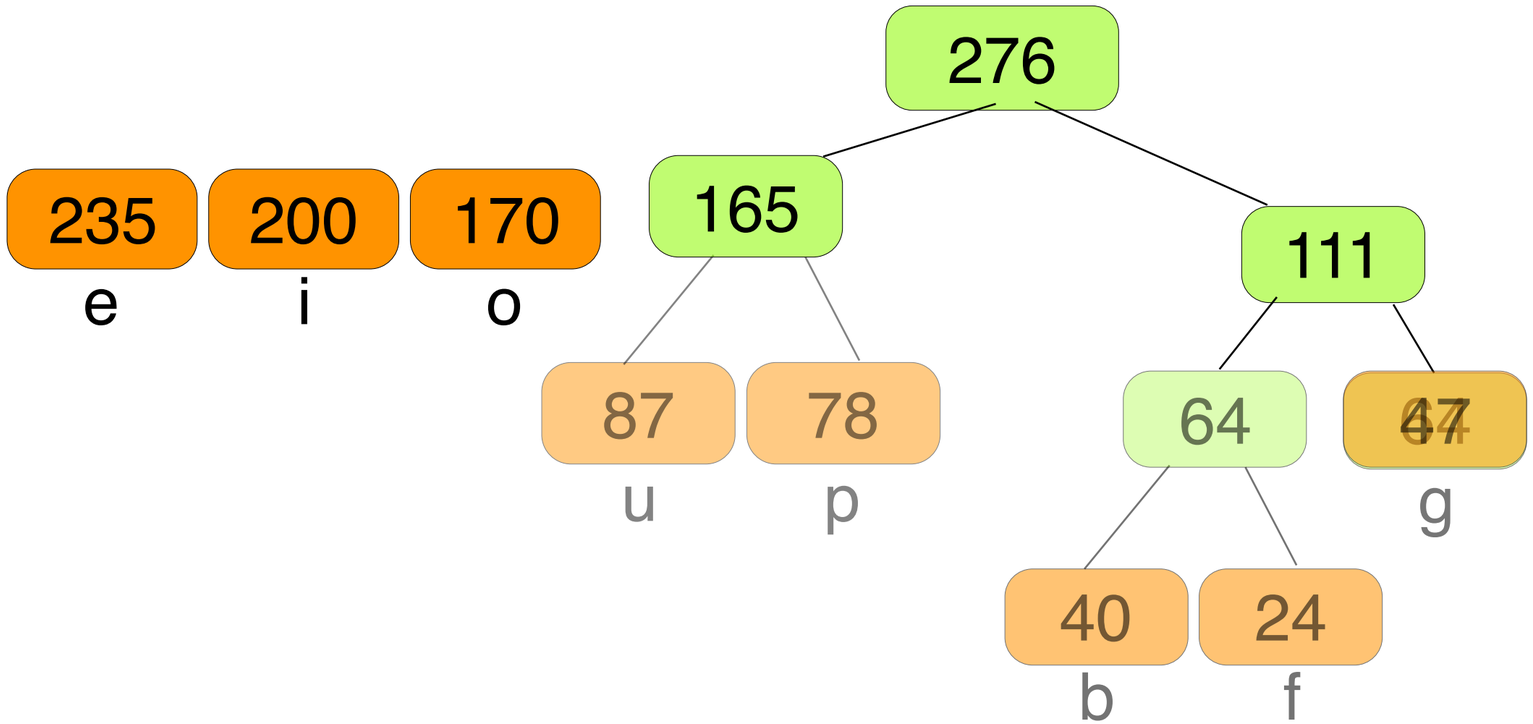

235
200
170
87
78
64
47
e
i
o
u
p
g
40
24
b
f

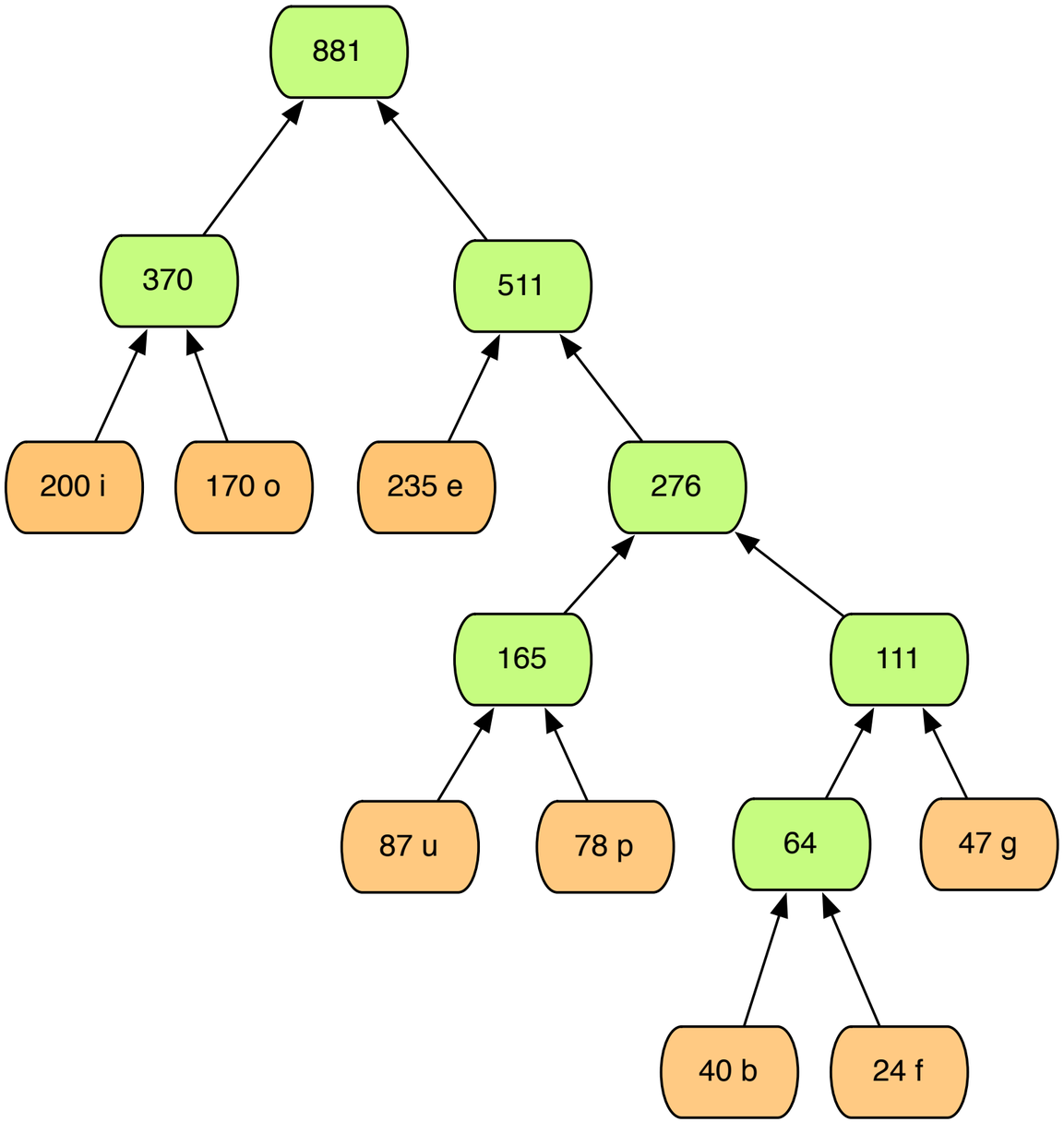

111
235
200
170
87
78
64
47
e
i
o
u
p
g
40
24
b
f

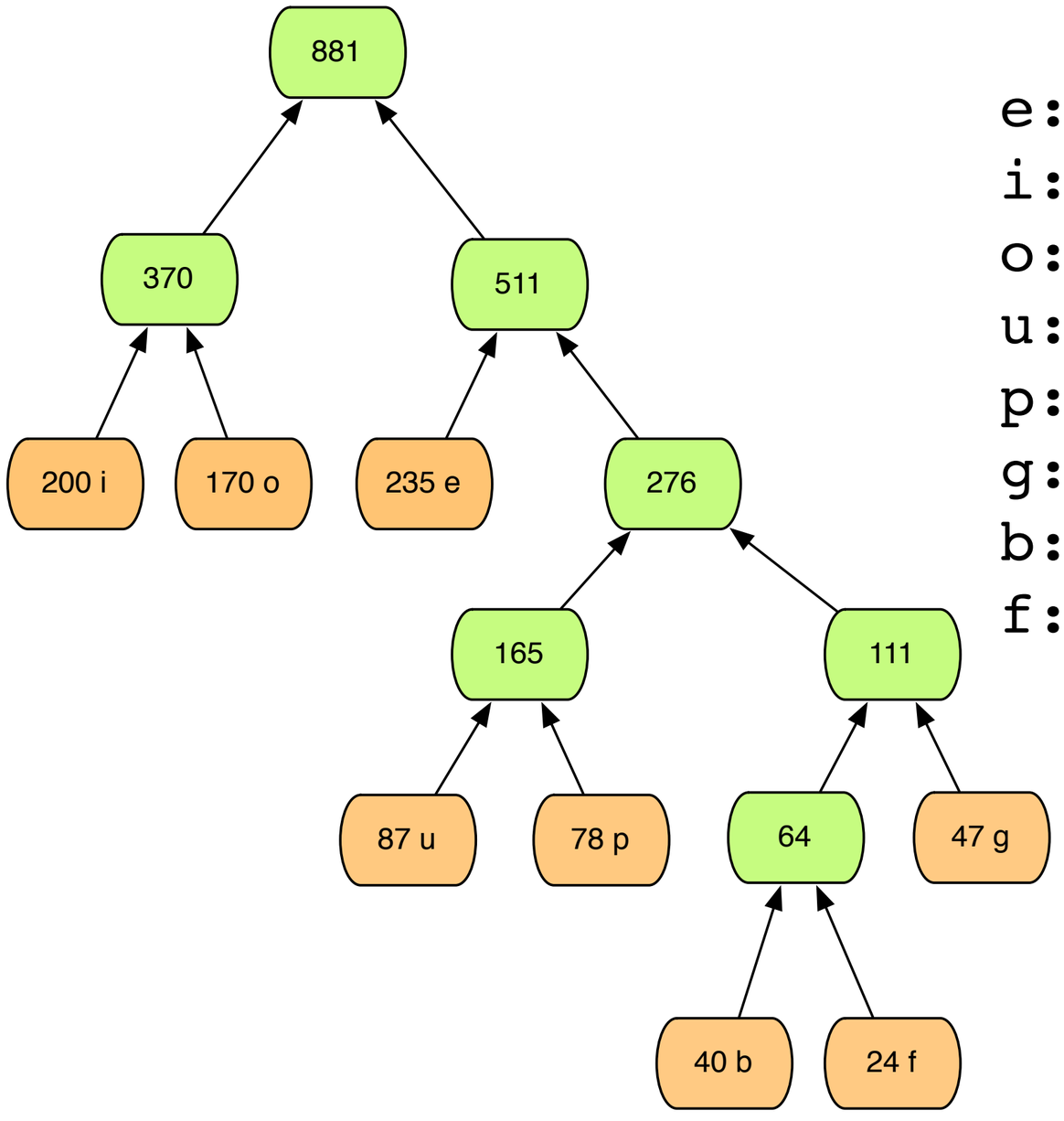

235
200
170
87
78
111
e
i
o
u
p
64
47
g
40
24
b
f


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

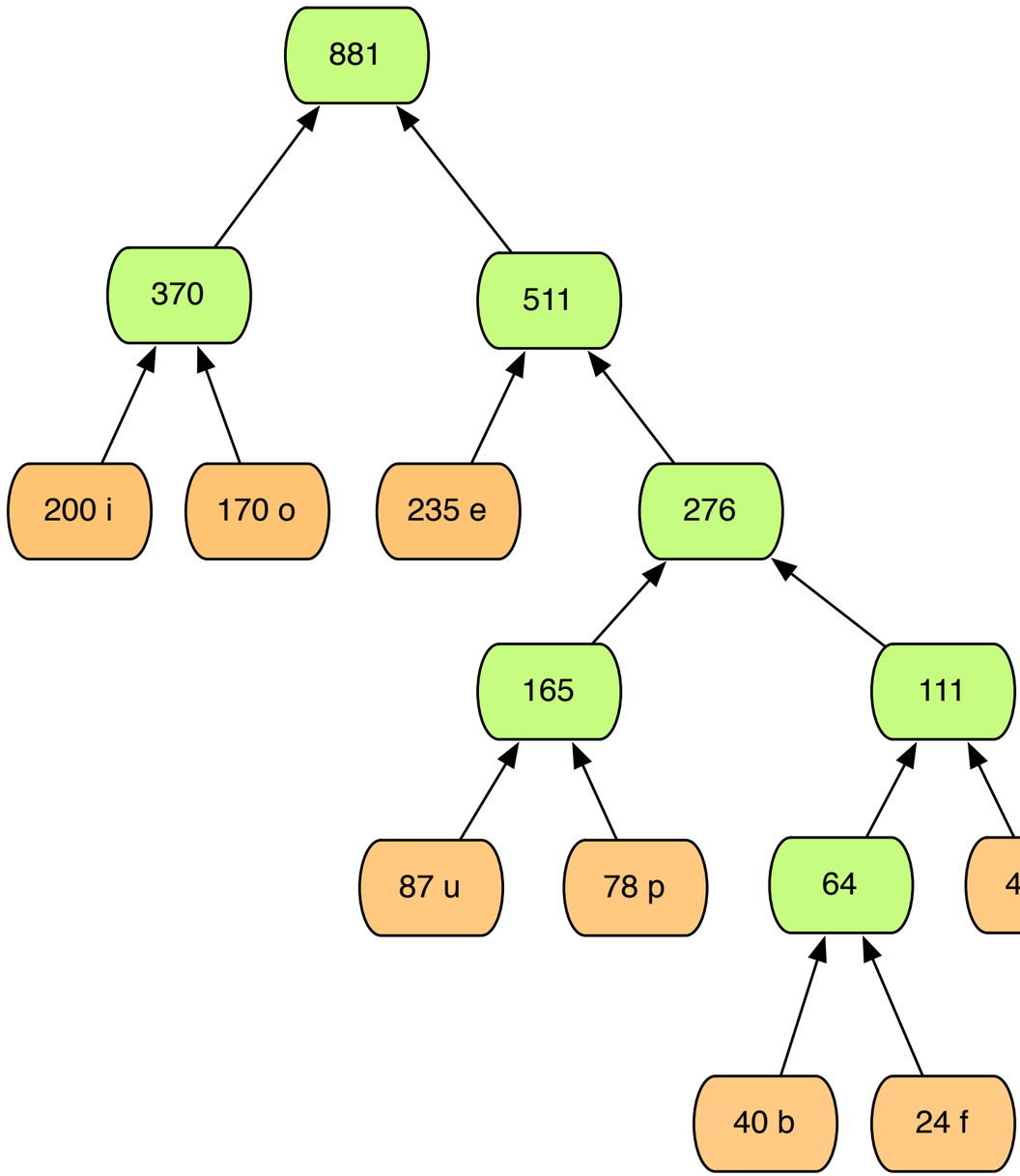

165
235
200
170
111
87
78
e
i
o
u
p
g
b
f


276
235
e
200
i
170
o
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
```

compare with slide I,

2378 vs 2643.

objective

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

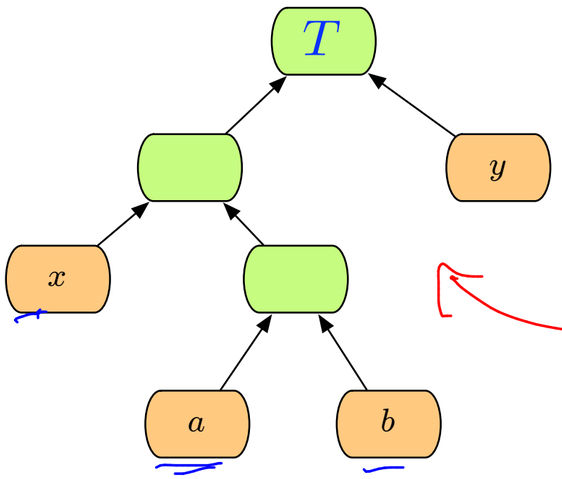
Prove the resulting tree is optimal.

exchange argument

lemma: Let f_x and f_y be the symbols with the smallest frequencies. There exists an optimal tree in which 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.



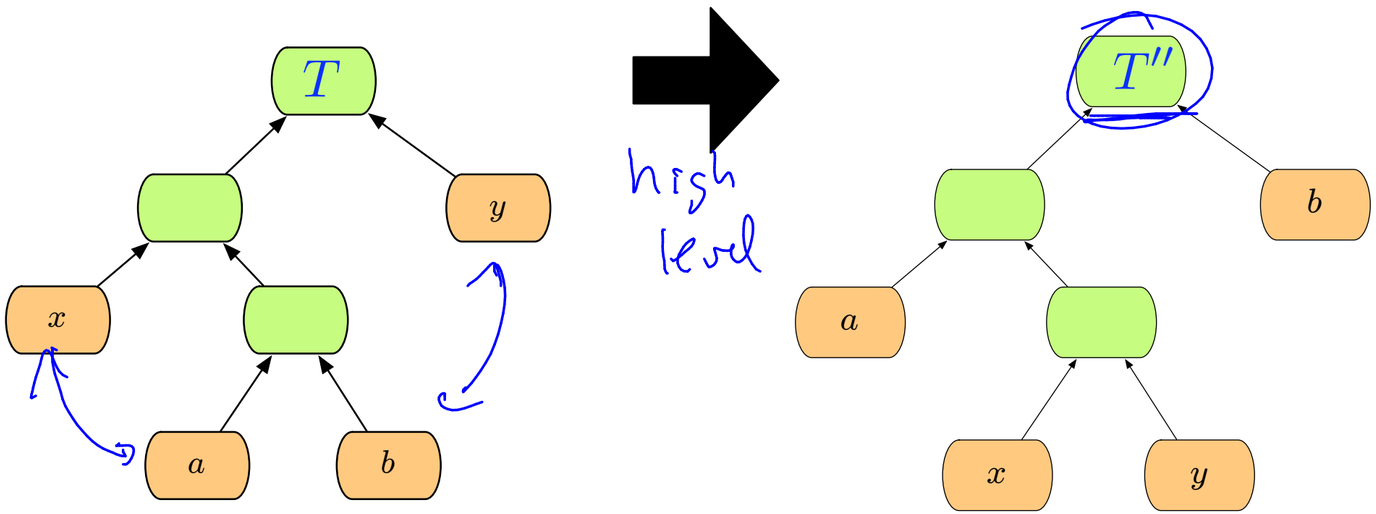
Proof: Let T be an opt. p.f. code for C.

If x,y are siblings, then the lemma holds.

Else they are not. Let a,b be the 2 characters of lowest depth that are siblings. *

why do a,b exist?? Ans: b/c the 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 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, f_y have the smallest frequencies.

wlog

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

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

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

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

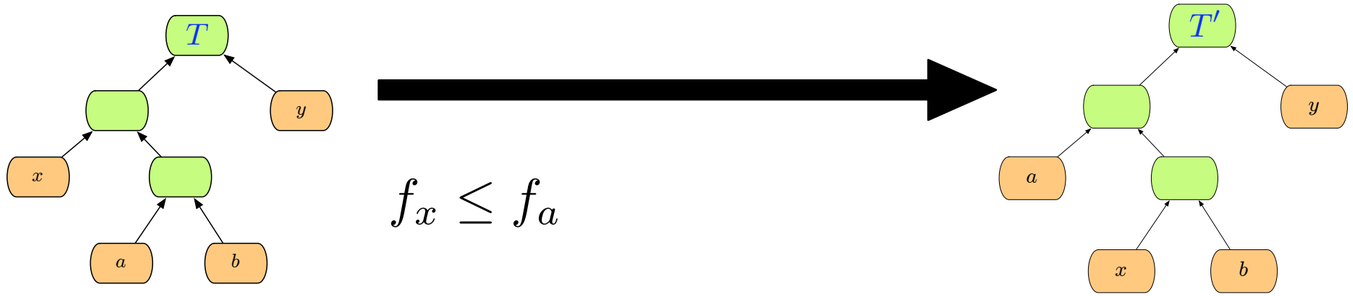
$= f_x(l_x - l_a) + f_a(l_a - l_x) = (l_a - l_x)(f_a - f_x)$

$(-1)(l_a - l_x)$

≥ 0 $\quad \geq 0$

≥ 0

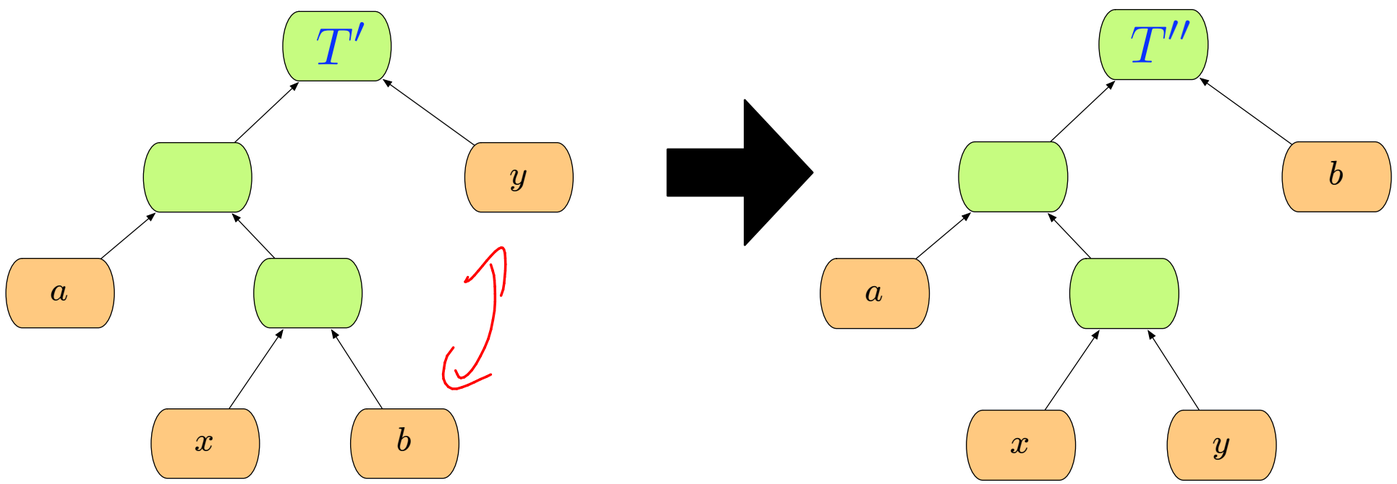
T is optimal $\Rightarrow$ difference must be $0 \Rightarrow T'$ is optimal.



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

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

exchange argument

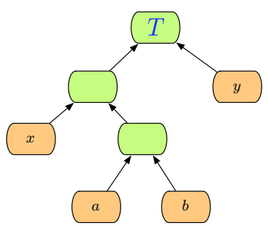


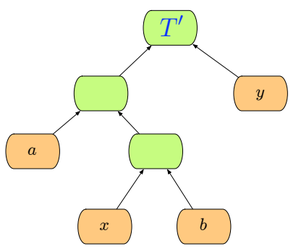
$$B(T') - B(T'') \geq 0$$

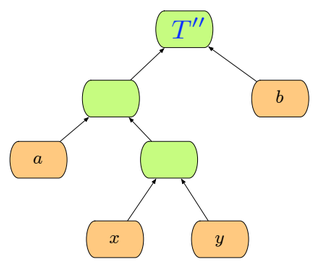
do the same argument with y and b.

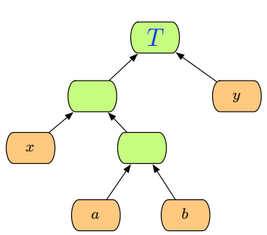
$\Rightarrow$ T'' is optimal, & it has x, y as siblings.

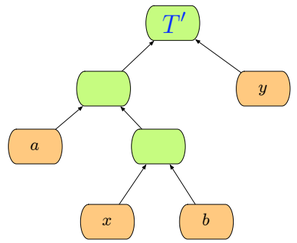
$\square$

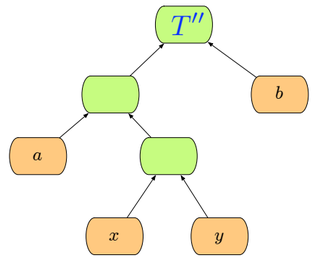

T
y
x
a
b


T'
y
a
x
b


T''
b
a
x
y





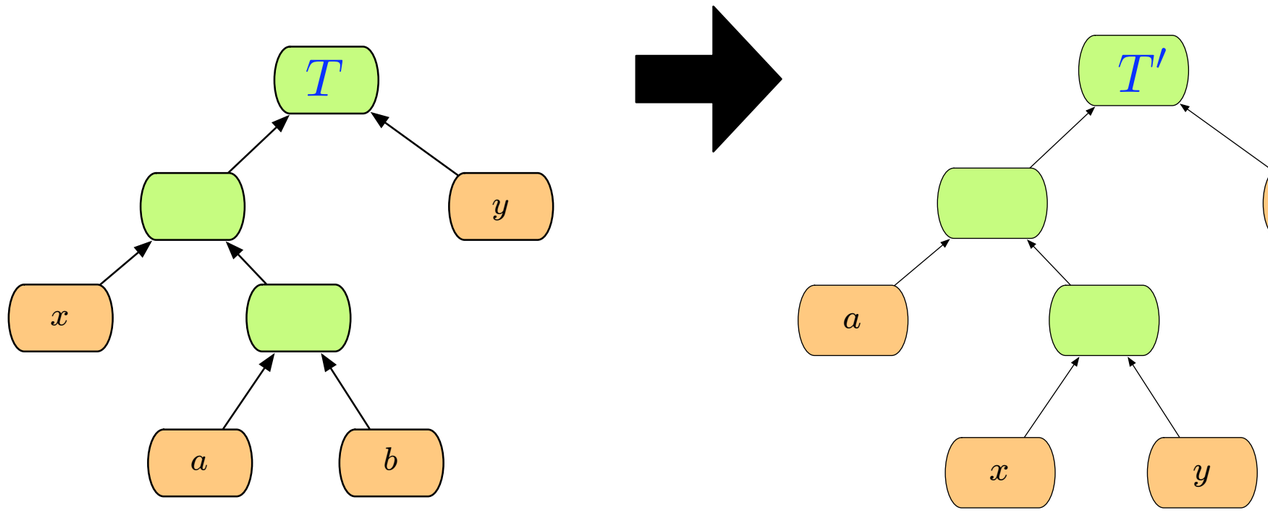


$$B(T) - B(T') \geq 0 \qquad B(T') - B(T'') \geq 0$$

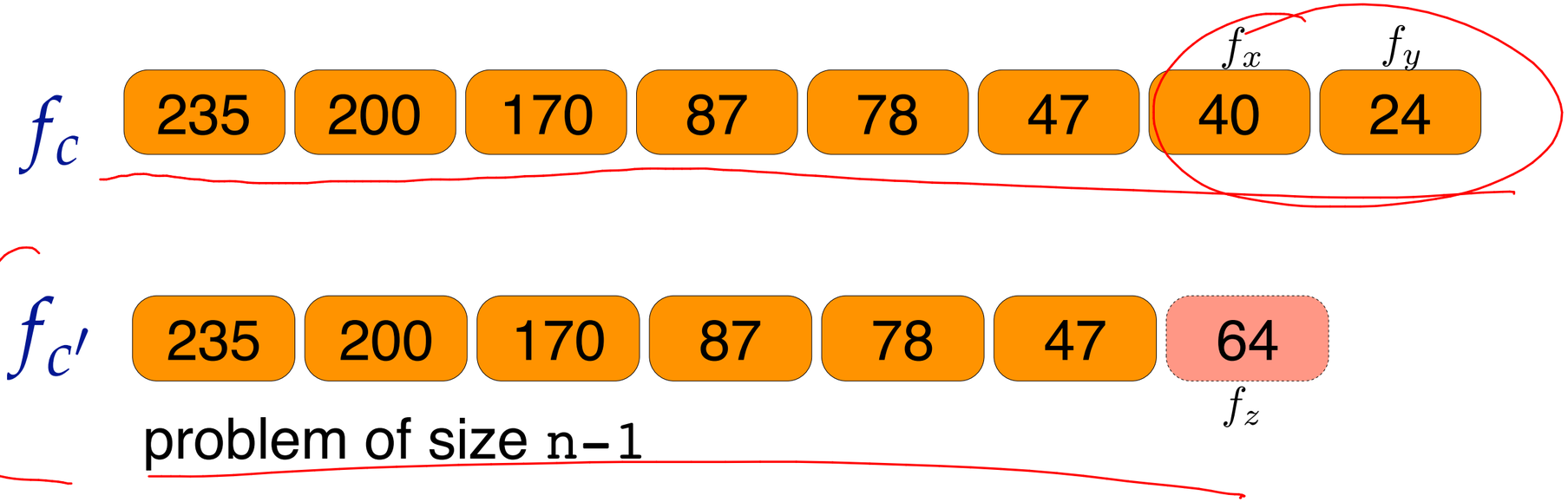
T'' is also optimal

exchange argument

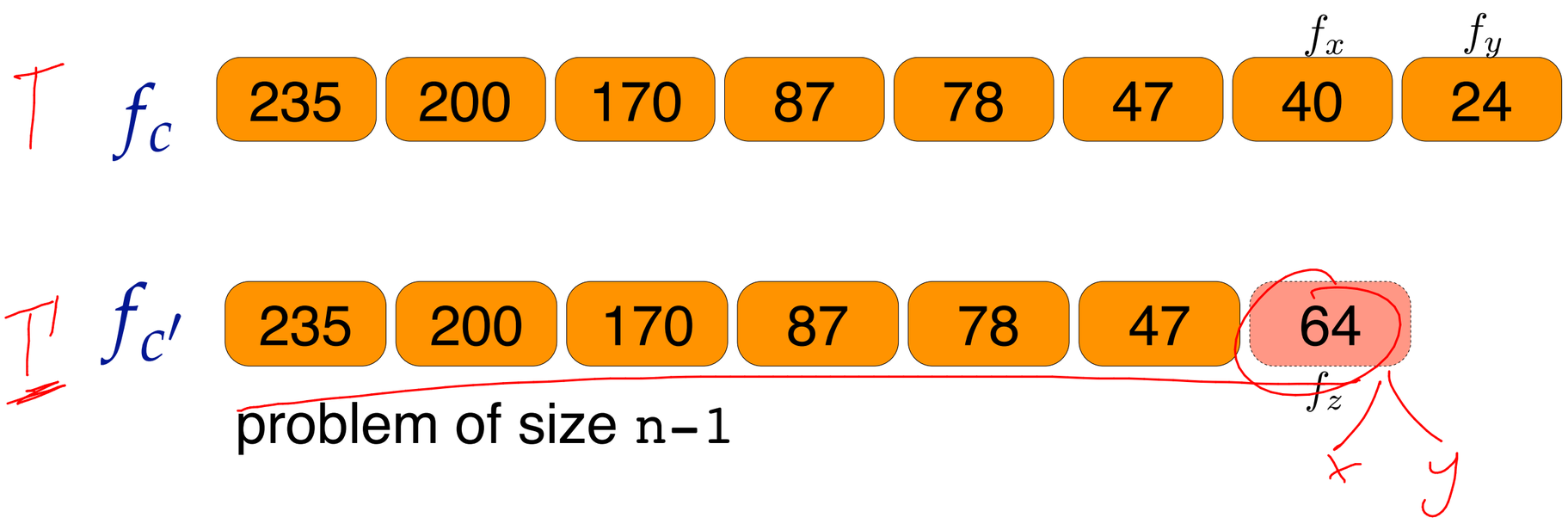
LEMMA: Let $x, y \in C$ be characters with smallest frequencies f_x, f_y. There exists an optimal prefix code T'' for C in which x, y are siblings. That is, the codes for x, y have the same length and only differ in the last bit.



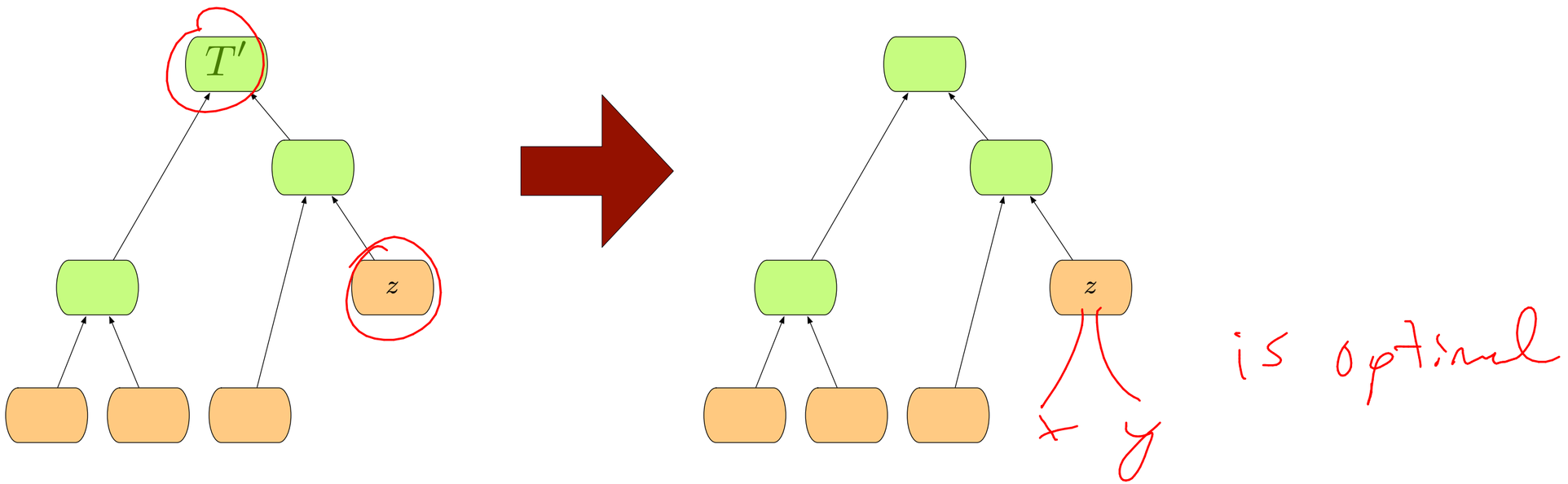
optimal sub-structure

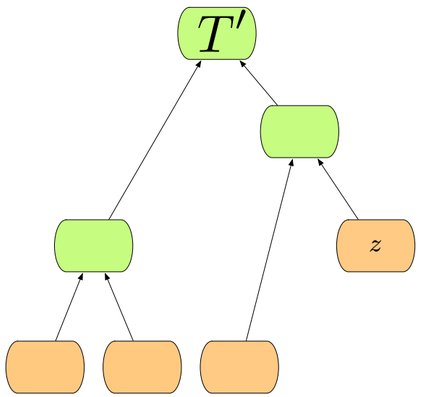


optimal sub-structure

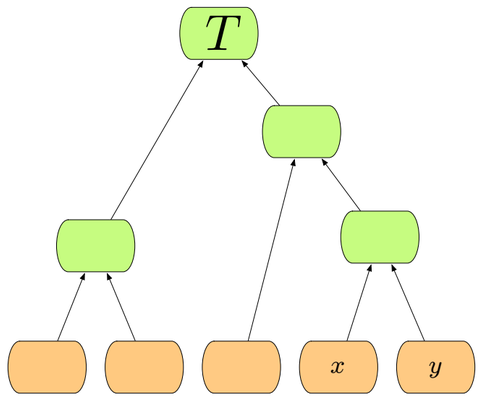


Lemma: The optimal solution for T consists of computing an optimal solution for T' and replacing the left z with a node having children x, y.


T'
z
z
x
y
is optimal



$B(T')$

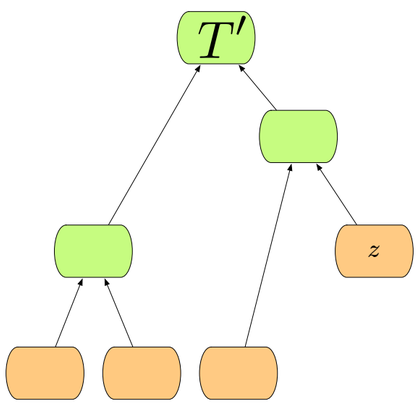


$B(T)$

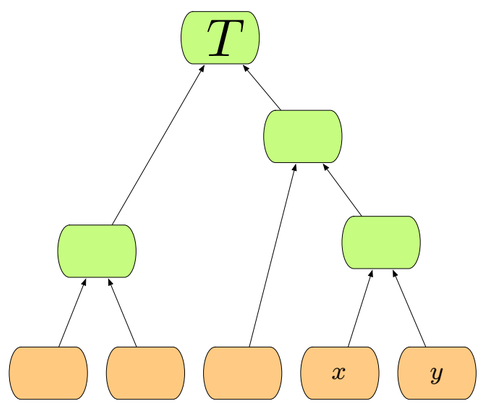
$l_x = l_z + 1$

$l_y = l_z + 1$

$B(T) = B(T) - f_x - f_y$



$B(T')$



$B(T)$

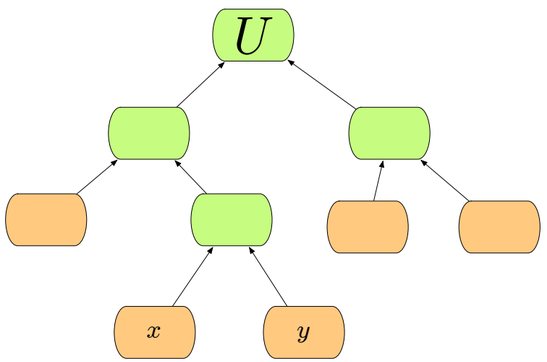
$$B(T') = B(T) - f_x - f_y$$

Suppose T is not optimal.

I.e. $\exists$ another tree U s.t. $B(U) < B(T)$

Suppose T is not optimal.

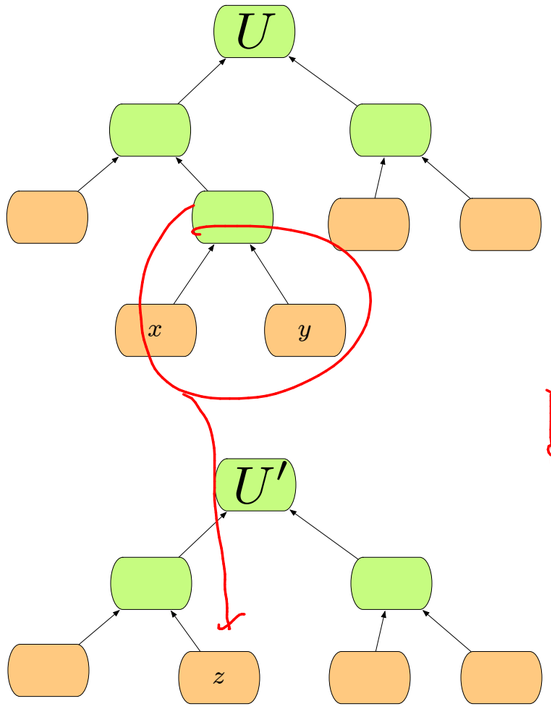
i.e. $\exists$ U s.t.



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

wlog, we can assume that x, y are siblings in U by Lemma I.

Suppose T is not



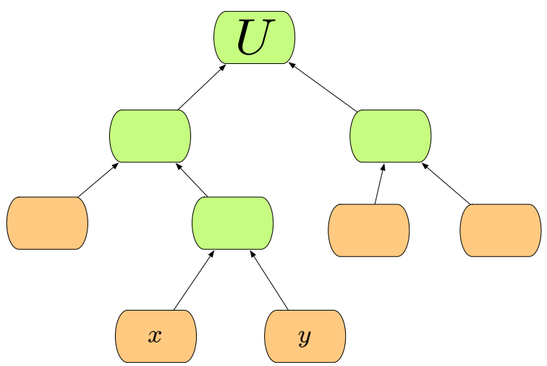
$$B(U) < B(T)$$

Define a tree U' s.t. x & y are combined.

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

contradicts the assumption that T' was optimal $\Rightarrow$ T must have also been optimal ☒

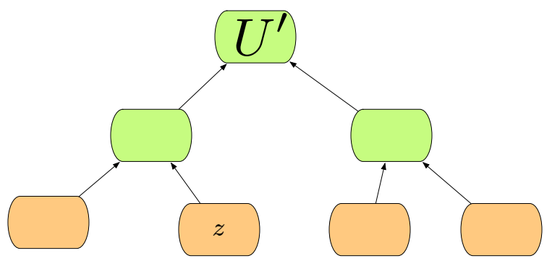
Suppose T is not



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

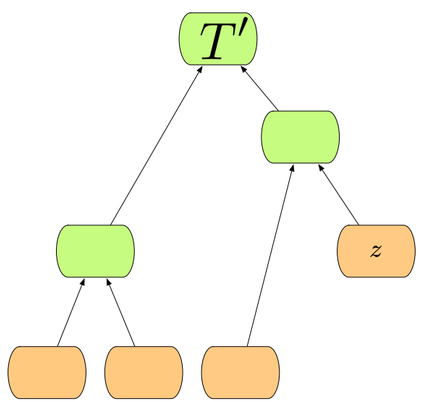
$$B(U') = B(U) - f_x - f_y$$

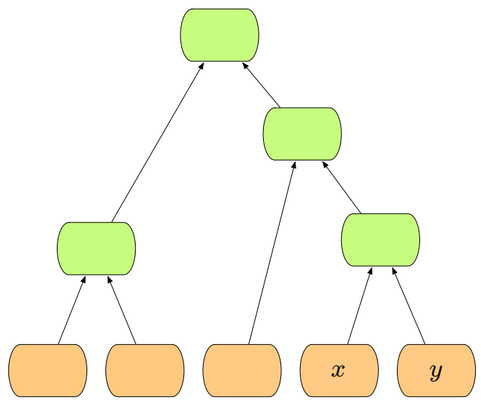
< B(t) - fx - fy



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

therefore





summary of argument

MST

connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

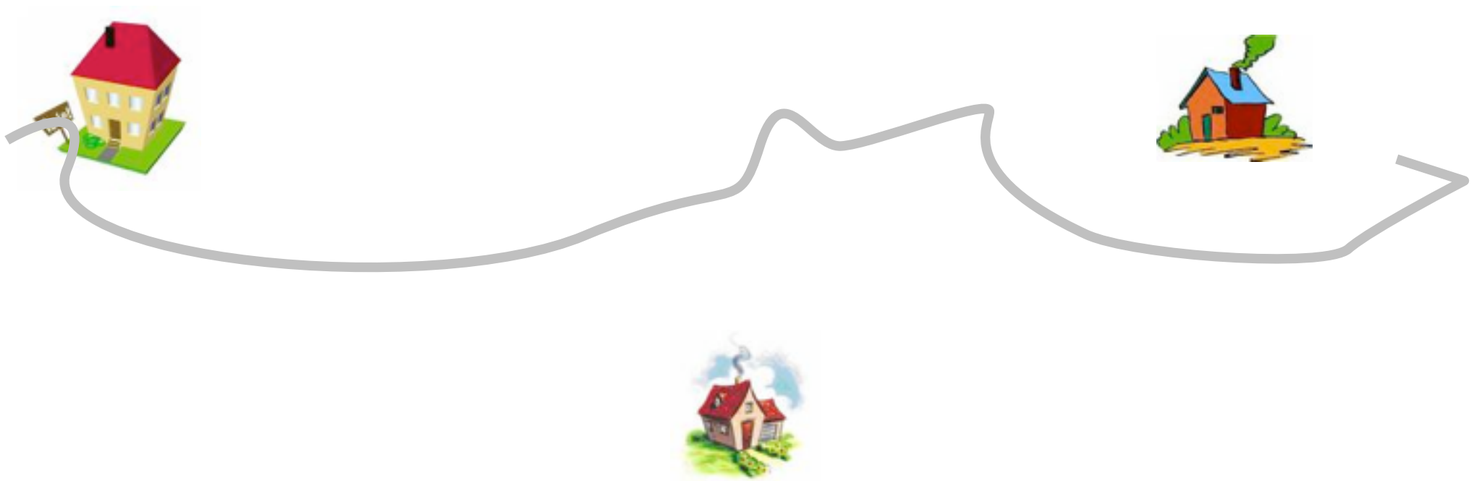


image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

connecting houses

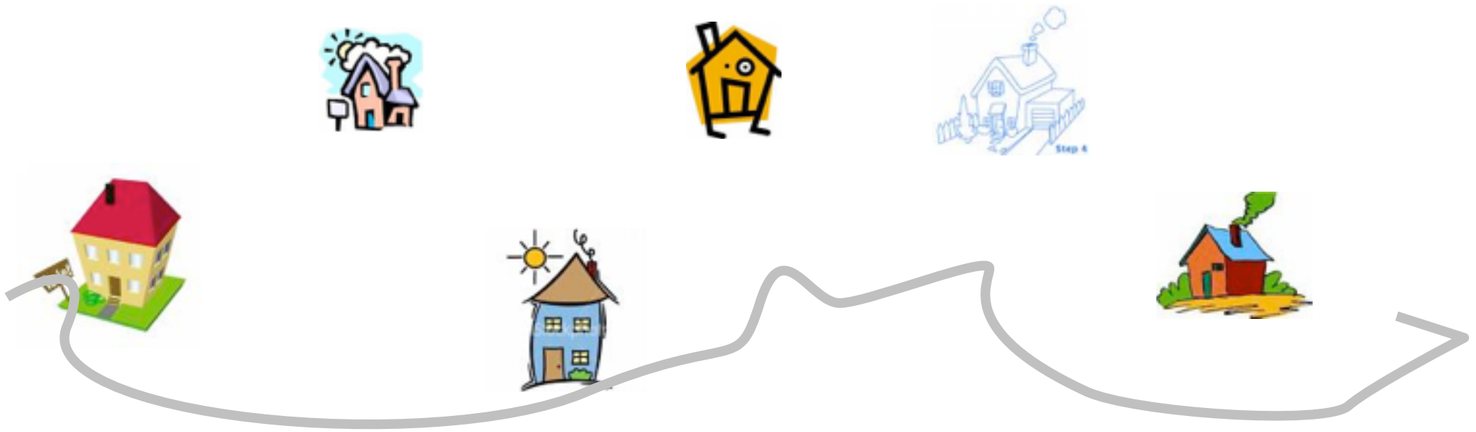
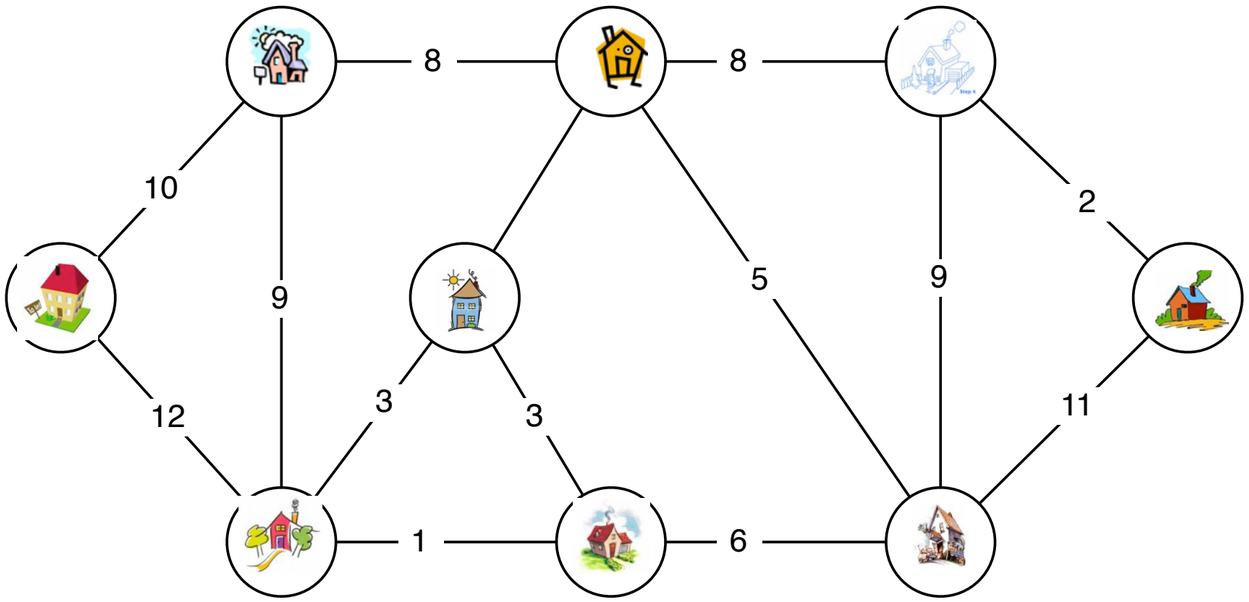


image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

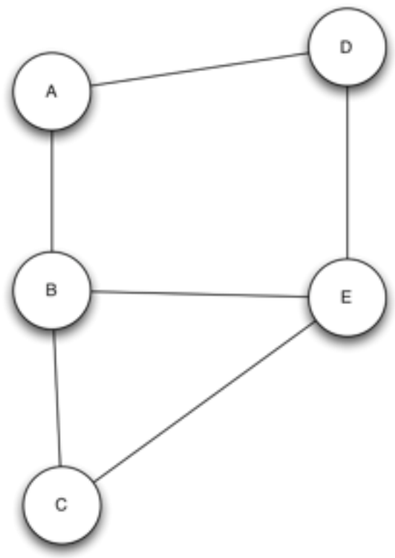
connecting houses

image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com


8
8
10
2
9
5
9
12
3
3
11
1
6

clrs [ch 22]

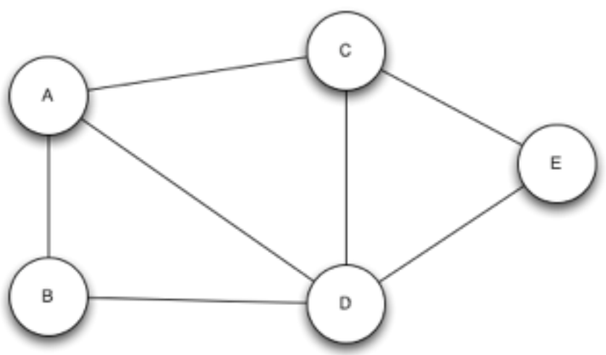
$$G = (V, E)$$



representation

$$G = (V, E)$$

adjacency list



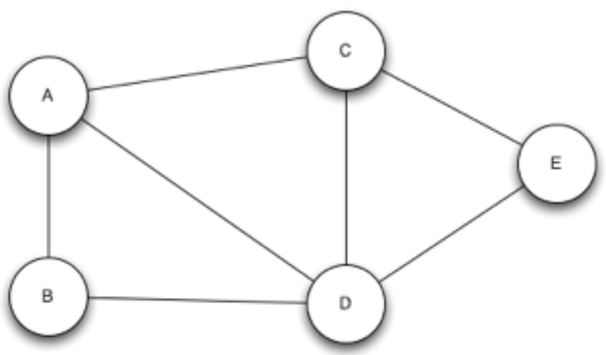
space:

time list neighbors:

time check an edge:

representation

$$G = (V, E)$$



adjacency matrix

space:

time list neighbors:

time check an edge:

definition: path

a sequence of nodes $v_1, v_2, \ldots, v_k$

with the property that $(v_i, v_{i+1}) \in E$

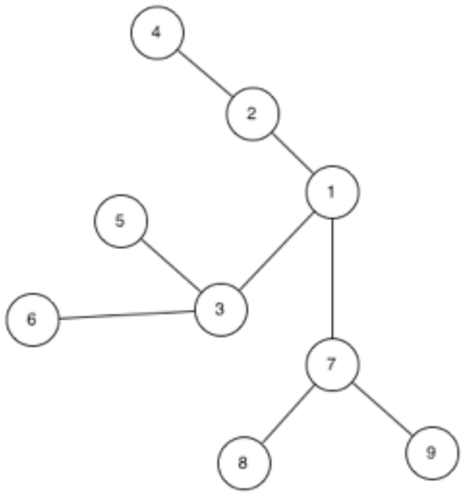
simple path:

cycle:

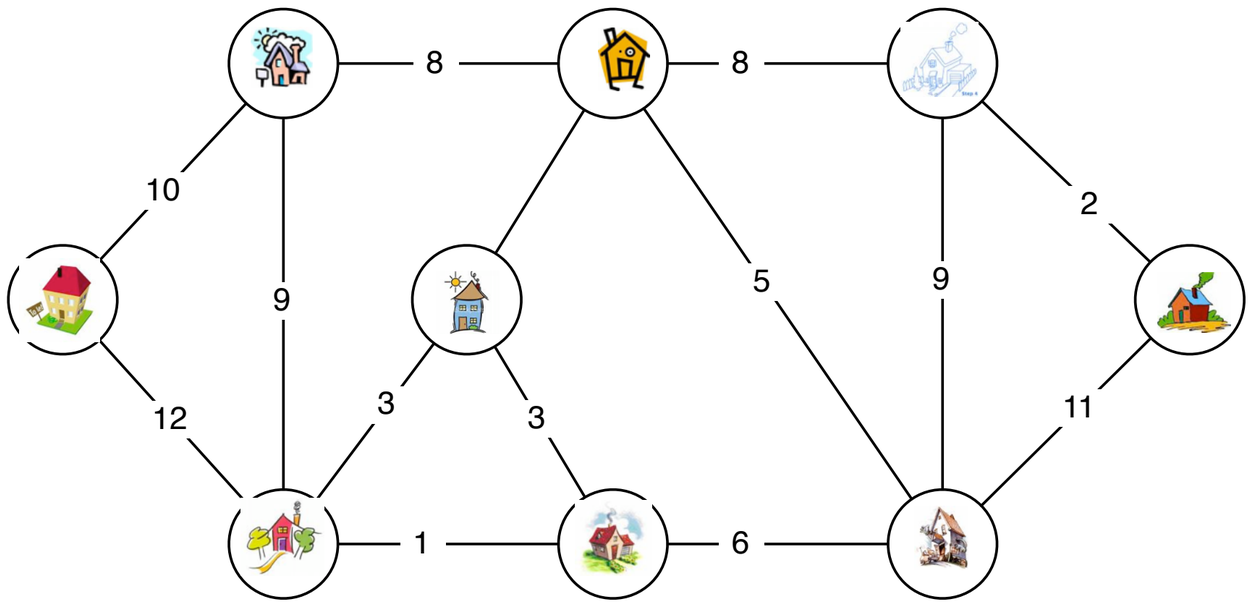
definition:tree

connected graph:

a tree is



what we want:



minimum spanning tree

looking for a set of edges $T \subseteq E$ that

(a) connects all vertices

(b) has the least cost

$$\min \sum_{(u,v) \in T} w(u,v)$$

facts

looking for a set of edges $T \subseteq E$ that

(a) connects all vertices

(b) has the least cost $\min \sum_{(u,v) \in T} w(u,v)$

how many edges does solution have ?

does solution have a cycle?

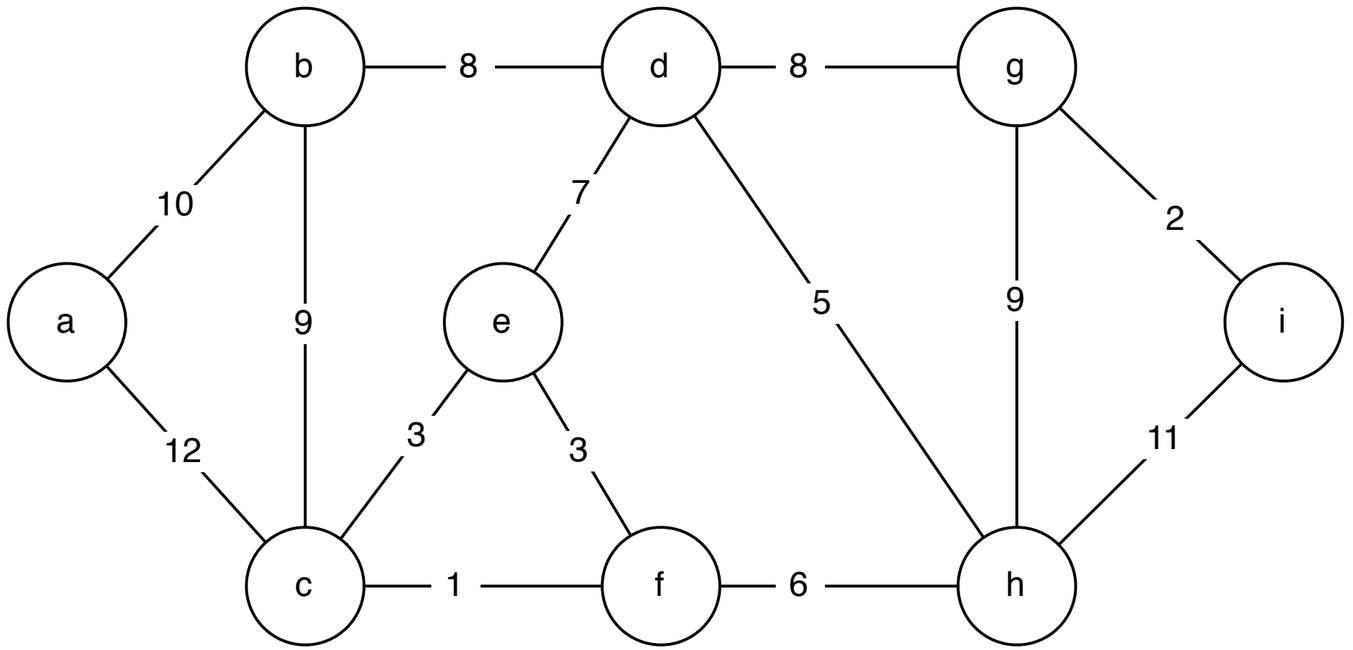
strategy

start with an empty set of edges A

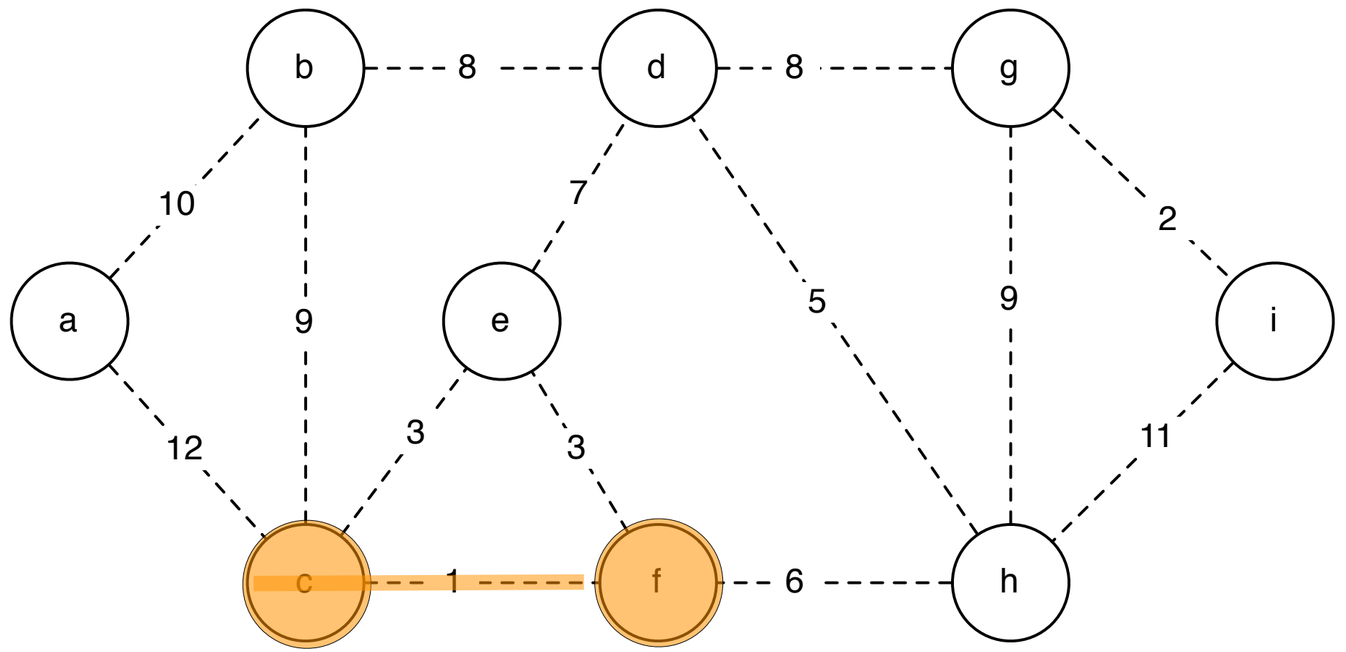
repeat for v-1 times:

add lightest edge that does not create a cycle

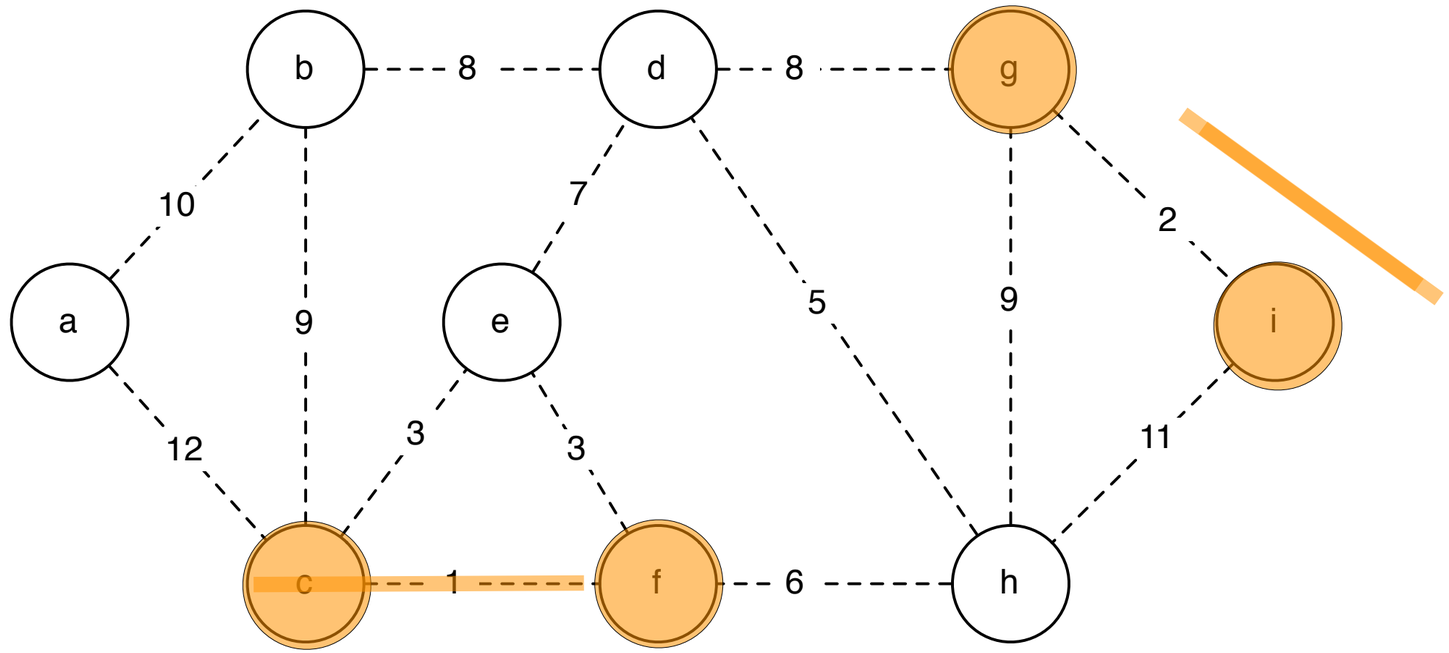
example



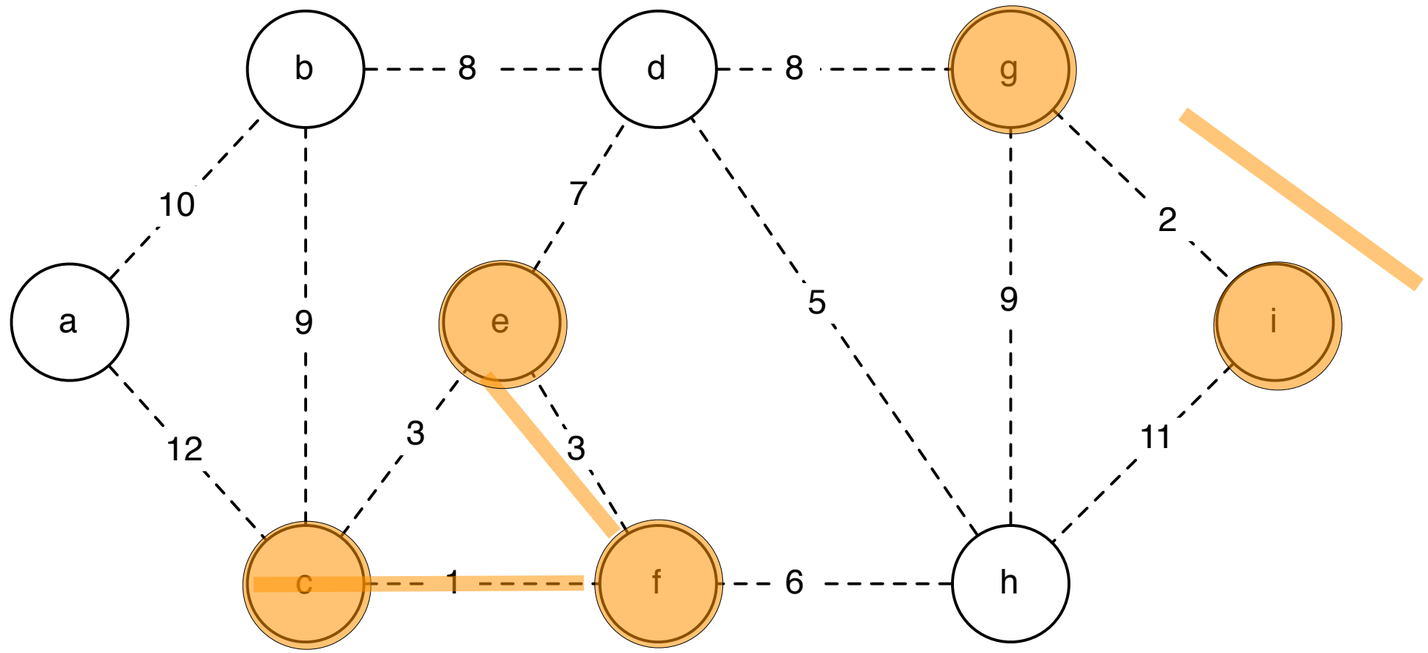
kruskal



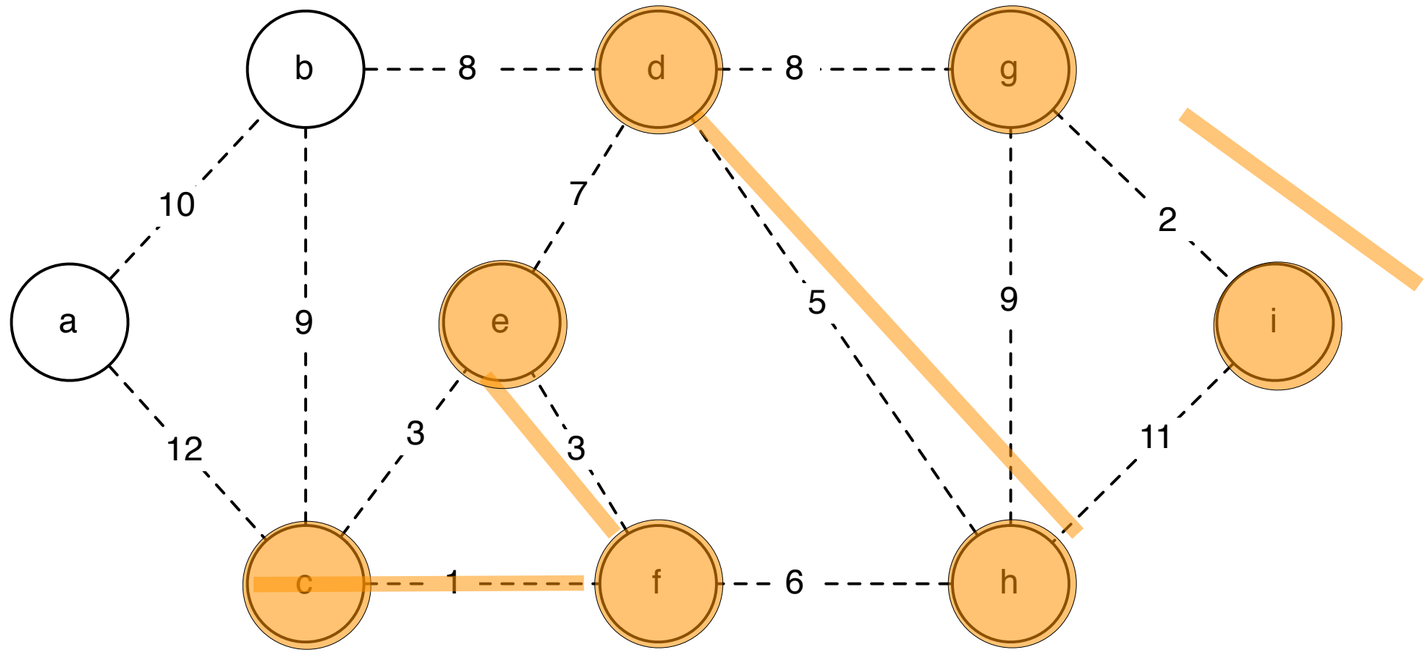
kruskal



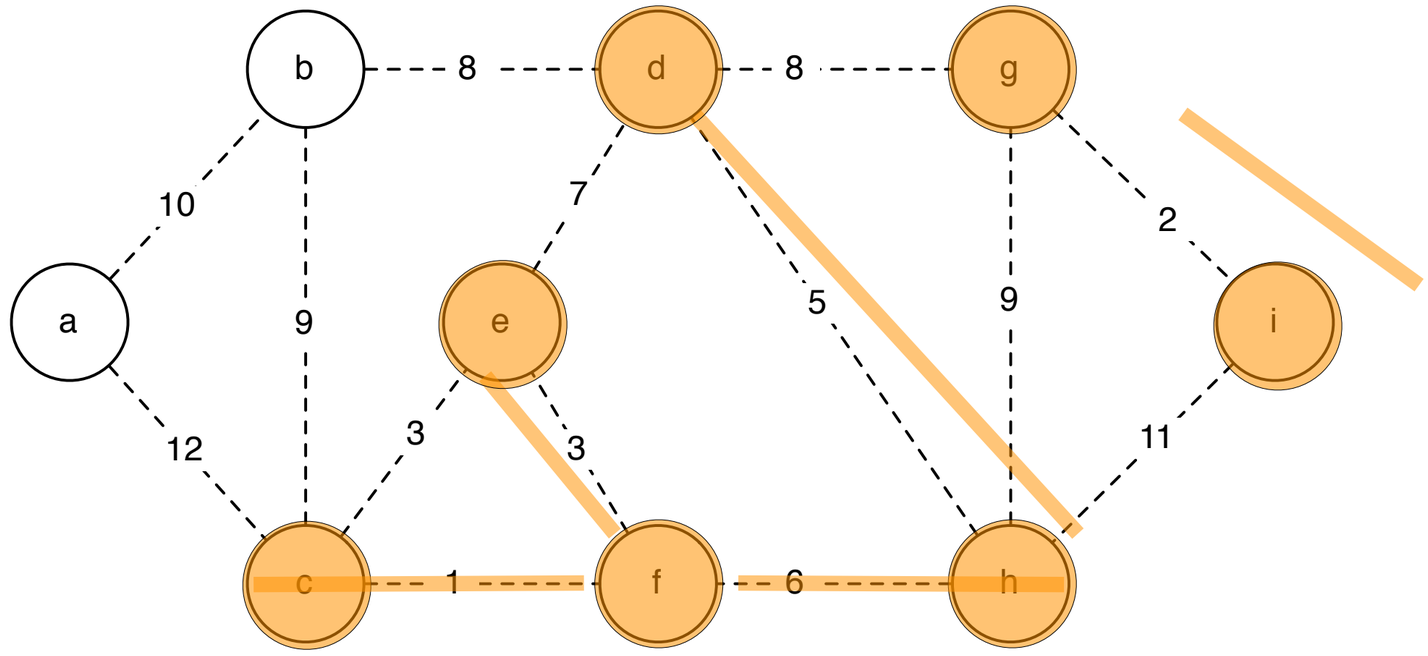
kruskal



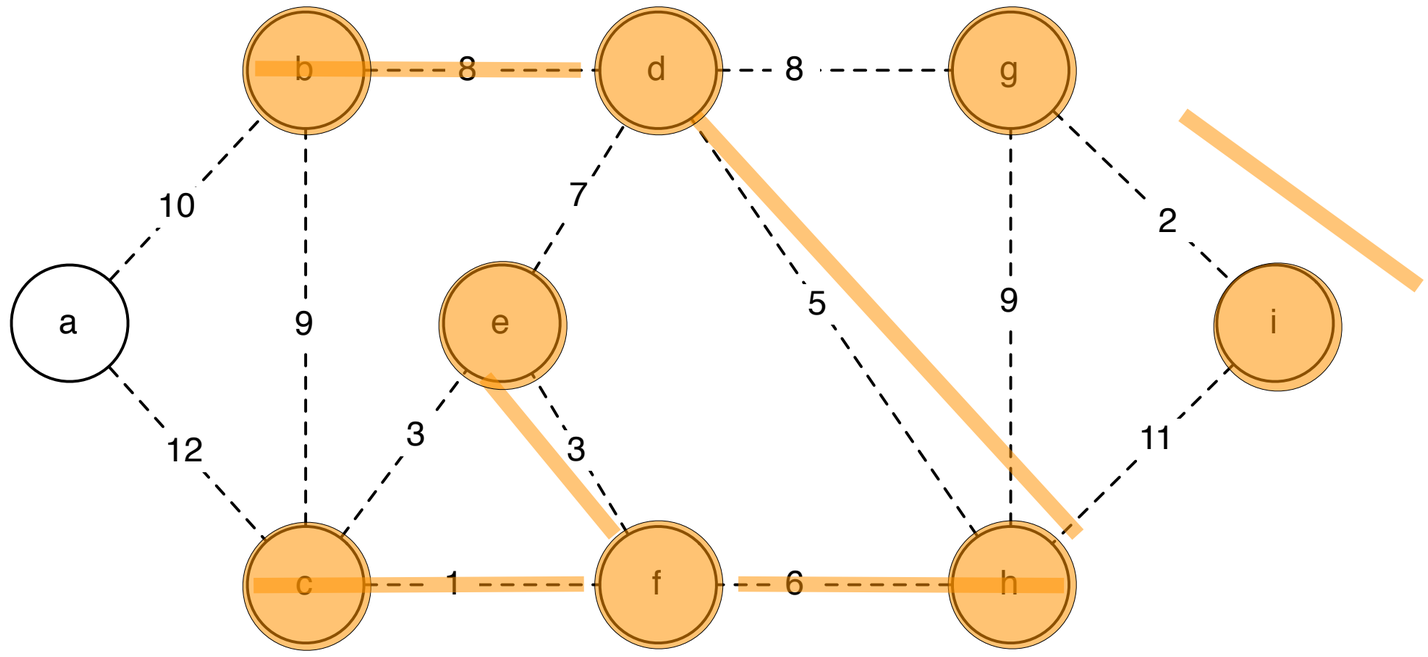
kruskal



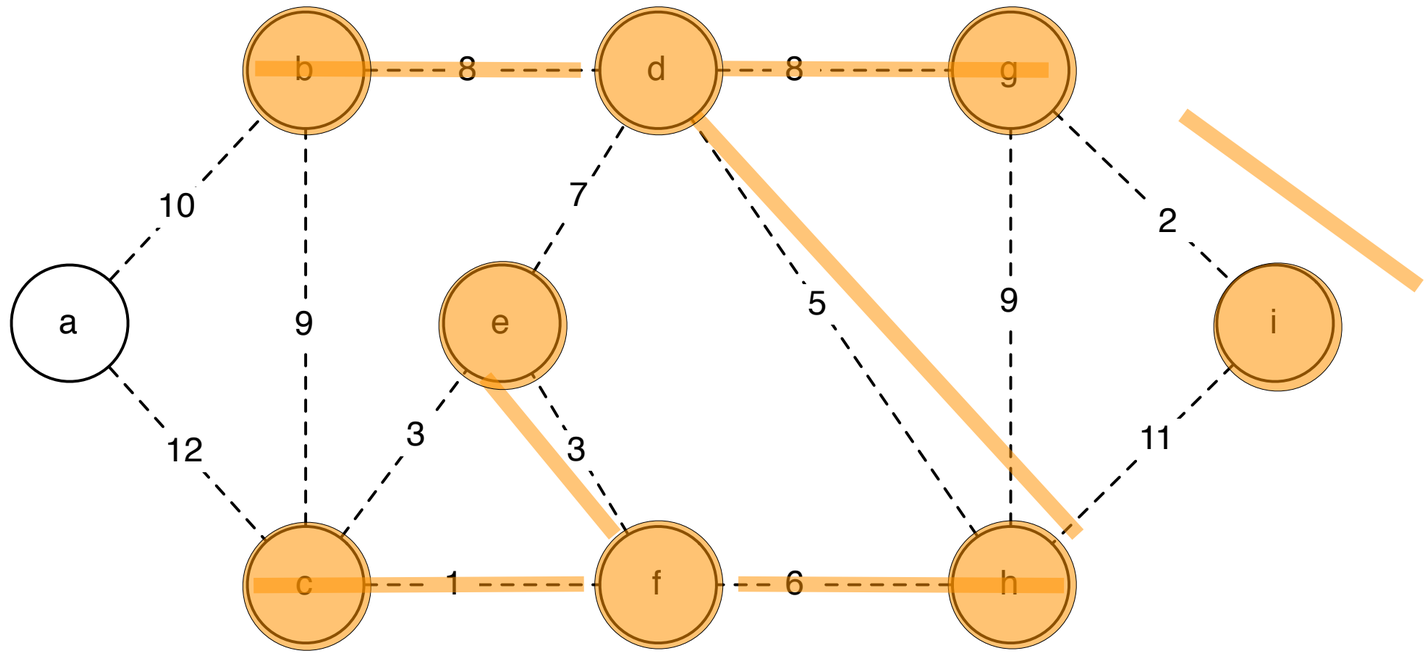
kruskal



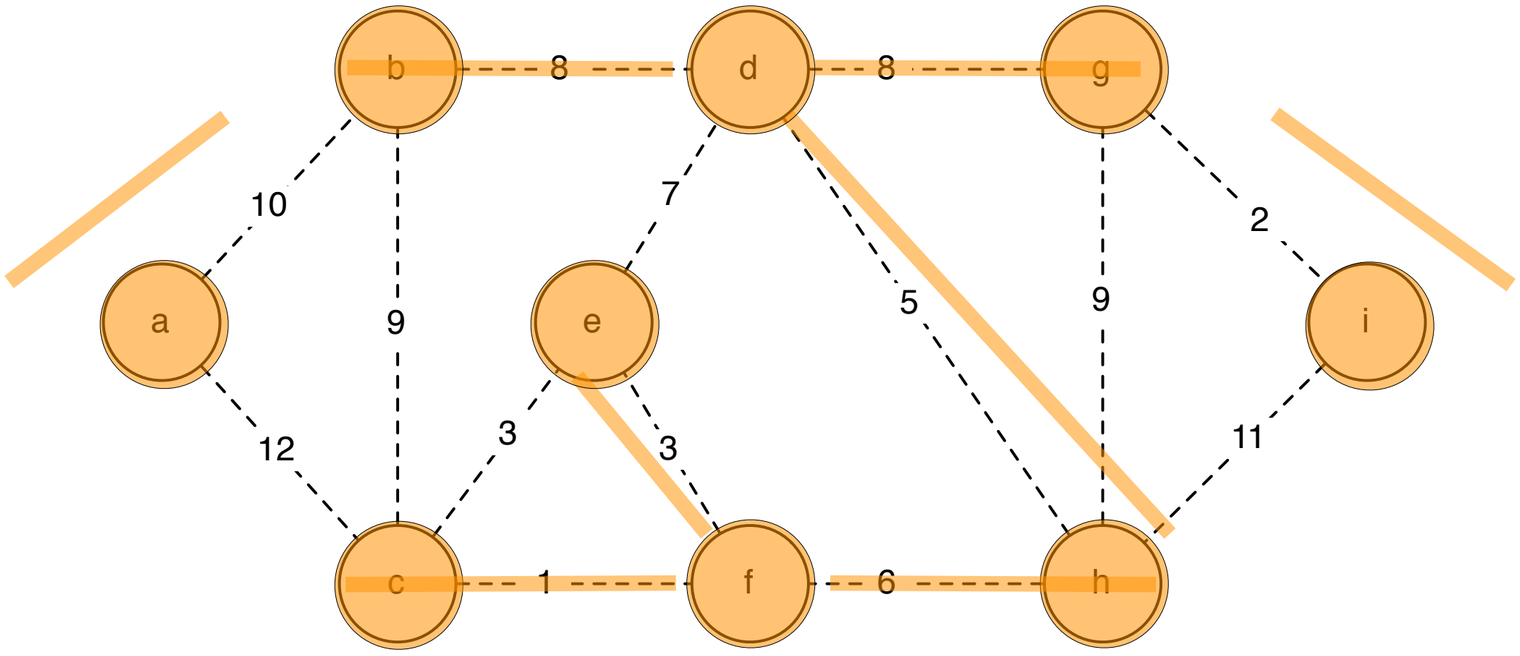
kruskal



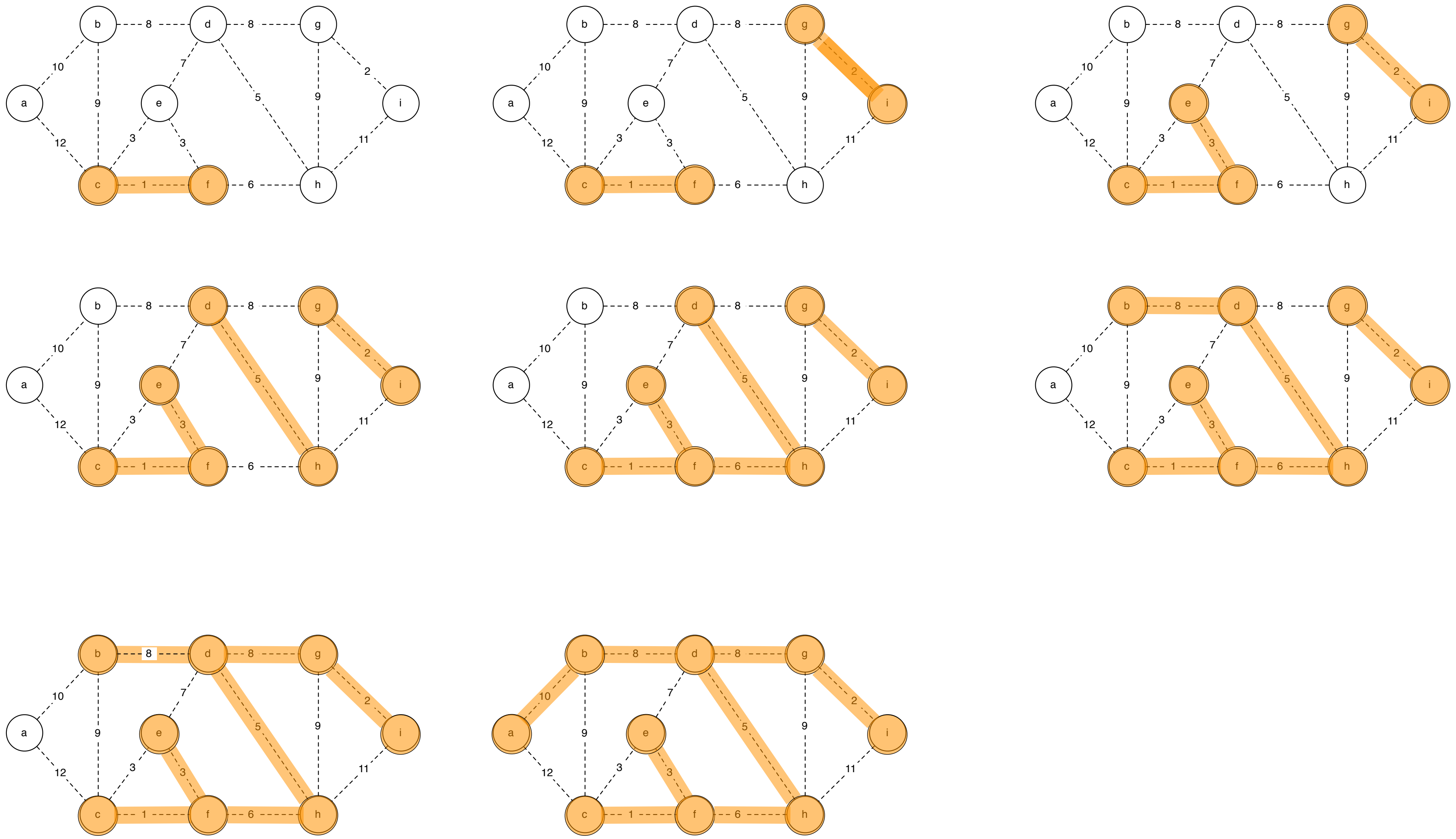
kruskal



kruskal



kruskal



why does this work?

1 $T \leftarrow \emptyset$
2 **repeat** $V - 1$ times:
3 add to T the lightest edge $e \in E$ that does not create a cycle